PHD

STEP-NC Enabled Cross-Technology Interoperability for CNC Machining

Safaieh, Mehrdad

Award date:
2014

Awarding institution:
University of Bath

STEP-NC Enabled Cross-Technology Interoperability for CNC Machining

Mehrdad Safaieh

A thesis submitted for the degree of Doctor of Philosophy

University of Bath

Department of Mechanical Engineering

20 December 2013

Table of Contents

Acknowledgments

This work documented in this thesis was carried out during the years 2009-2013 at the University of Bath. This journey would not have been possible without the valuable support of many people.

First and foremost I would like to thank my supervisor and friend, Dr Aydin Nassehi for giving me the opportunity to explore my research area and bringing me here to this fascinating city and university. I thank you sir, not only for your support during my academic career but also for your advice, which has changed my outlook on life. Without this advice, incisive questioning and inspired discussions I would not have reached my current level in my research.

I would also like to express my gratitude to Professor Stephen T. Newman for guiding me with such skill and good humour throughout my research work.

Thanks are also due to all my friends in the United Kingdom and Iran for making my life more interesting and more fun, and especially my friends in Bath who have helped me to enjoy my experience of research student life.

Last, but not least, I want to express my thanks to my parents for their support, without which none of this would be possible and, especially, my gratitude to my brother, Saman, who is to me more than a brother and friend.

Abstract

In recent decades there has been a rapid development of technology in manufacturing industries, in particular through the increasing use of ever more powerful and sophisticated Computer Numerical Controlled (CNC) machines to manufacture complex parts. These machines are supported by a chain of computer based software solutions amongst which manufacturing information is exchanged. With the need for information exchange, interoperability between various computer-aided systems (CAx) has become an important research area.

In CNC part programing, innovations by various hardware manufacturers and their reflection in their software have led to the necessity for the existence of different part programs for each machine. Creating these is a time consuming and economically inefficient activity. Implementing genuine interoperability between CNC machines is a way of eliminating this deficiency but, to achieve this, CNC programmers must be able to write a CNC program for a specific machine and effortlessly convert that program to work for other machines.

The aim of this research was to enable the exchange of CNC programmes across machines with different technologies and demonstrate this between a C-axis CNC turn-mill machine and a 4-axis CNC machining centre. This has been achieved by designing a cross-technology interoperability framework that is capable of supporting systems that can work with the different types of CNC machines. This framework is the core contribution to knowledge from this PhD research. In order to fully identify the context for the research, this thesis presents a review of existing literature on machining of turn-mill parts and interoperability for CNC manufacturing. This is followed by the specification and realisation of a novel framework for cross-technology interoperability for CNC manufacturing. The demonstration is conducted using test components that can be manufactured using different CNC technologies.

List of Abbreviations

AIC	Application Interpreted Constructs
AIM	Application Interpreted Model
AM	Application Modules
APT	Automatically Programmed Tool
ARM	Application Reference Models
ATS	Abstract Test Suites
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CAx	Computer Aided systems
CNC	Computer Numerical Control
EDM	Electrical Discharge Machining
G&M	G codes and M codes
IGES	Initial Graphics Exchange Specification
IMS	Institute of Management Services
ISO	International Organisation of Standardization
NC	Numerical Control
OSI	Open System Interconnection
PC	Personal Computer
PLM	Product Life-Cycle Management
PPP	Point to Point Protocol
PPTP	Point to Point Tunnelling Protocol
SET	System d'Echange et de Transfer (System of exchange and transfer)
SSL	Secure Sockets Layer
STEP	The Standard for the Exchange of Product Data Model
TCP	Transmission Control Protocol
UML	Unified Modelling Language
UPD	User Datagram Protocol
VDA-FS	Verband der Automobilindustrie - Flächenschnittstelle (Organisation of the automotive industry - surface translation format)
XTSys	Cross-technology CNC interoperability system

List of Figures

List of Tables

1 Introduction

In the latter half of the 19th century, the manual and skill based approach to manufacturing was increasingly replaced by automated mass production techniques as the technology made available through the industrial revolution was utilised to achieve a faster pace of production at a lower cost (Kalpakjian and Schmid, 1992). This process continued in the 20th century, with a commonly cited example being the introduction of transfer machine lines. In the 1970s and ’80s increasingly flexible manufacturing methods were developed to enable the automated manufacturing of a wide variety of parts (Gunasekaran et al., 1995). An important element in these manufacturing methods was the emergence of Computer Numerical Controlled (CNC) machines, which were introduced in the late 1970s and which, not only enabled machine tools to be controlled with computers, but also had the capability of being reprogrammed for flexible production (Xu and Newman 2006). CNC machines with multi-axis and multi-process workstation configurations have now been developed to support high-speed manufacturing and the manufacture of complex products such as aircraft components (Newman et al. 2008).

Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) systems have come to be used as part of the CNC manufacturing process to enable the engineers to programme complex parts for manufacture using 2-dimensional drafting and 3-dimensional solid and surface modelling software tools have also become common techniques to drive CAD/CAM processes in manufacturing, especially where complex parts are concerned (Hou, 2008).

During the 1990s, more sophisticated CNC machines with PC-CNC controllers became the common tools for low tolerance, high accuracy mass manufacturing of a variety of parts. With the increasing level of complexity, sharing information between different machines became more difficult. This has been also due to lack of means for passing high-level information between CNC machines and, perhaps more importantly, lack of standardised methods for capturing data about their different capabilities and structures in a computer readable format (Newman and Nassehi, 2007).

Traditionally, one method of increasing possibilities in production has been to manufacture a component using a variety of machine tools and processes. In CNC manufacturing, however, this approach is prone to failure, with a likely cause of failure being deficiencies in the communication between CNC machines and their inability to use information from another vendor's machine due to the incompatibility of programming languages. Researchers are actively pursuing the goal of finding efficient, robust and reliable methods of communication between CNC machines to enable the sharing of information between different types of machines, the part of the process of exchanging information in the CAD/CAM/CNC chain entitled "interoperability in the CAx chain".

One of the core reasons for the difficulty in achieving interoperability is that the requirement for increased flexibility in the production of metal parts, combined with the development of a wide range of CNC machine tools, has resulted in a large number of different technologies and machine languages operating concurrently within a manufacturing process chain, with the consequence that full interoperability has become a highly complicated goal. Currently, a large number of NC coding system variations, standards, and languages are used to support the CAPP/CAM/CNC process chain. Figure 1.1, represents an ideal interoperable manufacturing network in which each CAx system is capable of exchanging information with any other CAx system in the adjacent links of the CAx network.

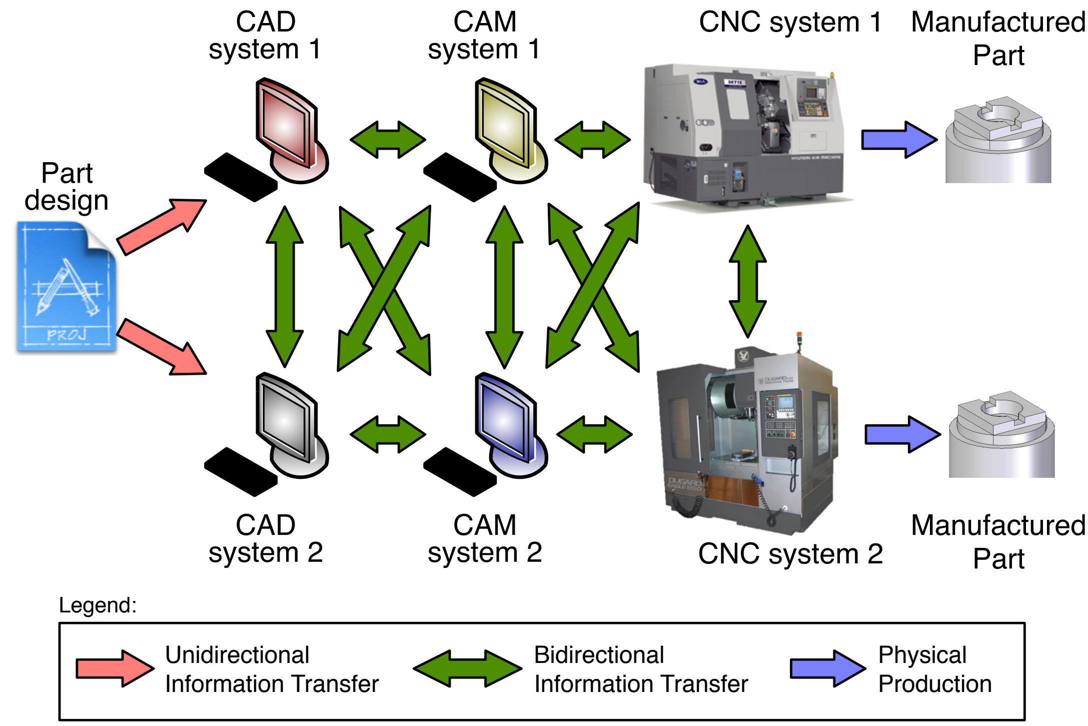


Figure 1.1: An interoperable CAx network
adapted from (Newman and Nassehi, 2007)

In an interoperable manufacturing system, interoperability between different CNC machines which use the same technology is relatively simple, because these machines use mainly the same features and the same operations to machine a part. The lack of a framework for generating CNC codes that can be interpreted by different CNC machine tools with different technologies, however, is a major obstacle to interoperability in CNC manufacturing because whilst machines with different technologies may use the same features they do not use the same operations to machine a part. Figure 1.2 illustrates this lack of interoperability between CNC machines with different technologies.

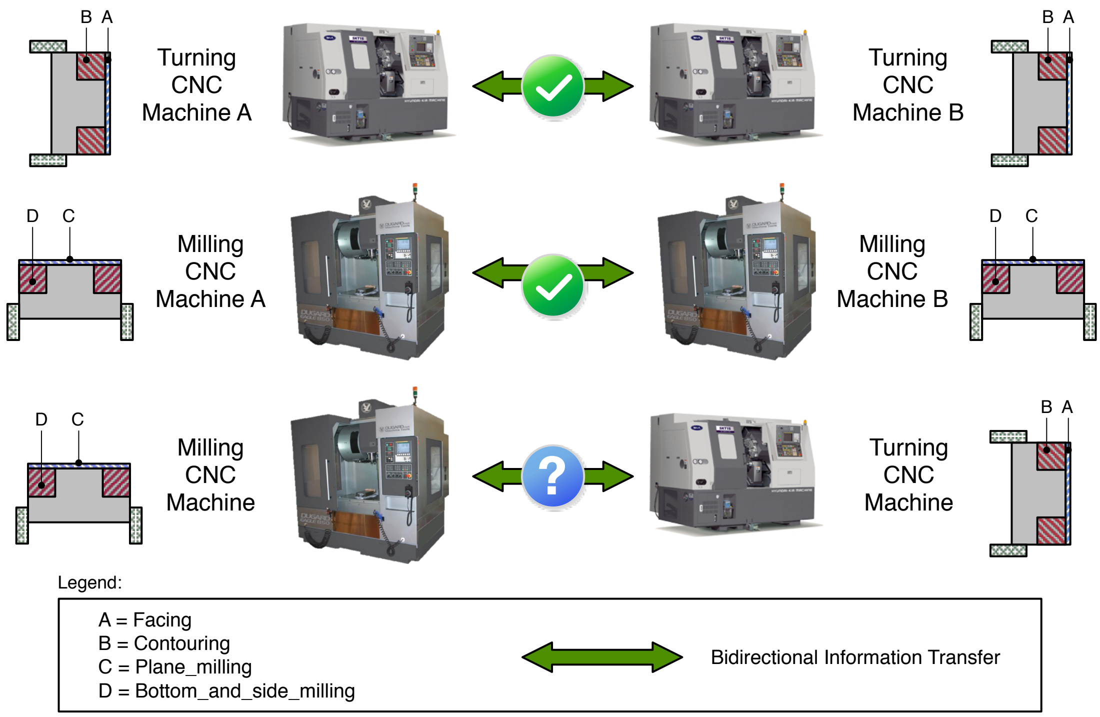


Figure 1.2: Interoperability between CNC machines with different technologies

It is in this context that this research aims to introduce an innovative approach to enable interoperability between CNC machine tools with different technologies, so that the same part may be produced on multiple machine types by converting one type of NC code to another. Through this approach it will be possible to convert codes between different machines without CAD/CAM files which may not exist for legacy parts; to machine a part using different technologies on the shop floor; to increase the rate of production by enabling the use of more machines for manufacturing a single part, and; to use different machines in cases where the preferred device has broken down.

To illustrate how a part might be manufactured using different machines on the shop floor, a 4-axis CNC milling machine and turn-mill centre will form the experimental basis for the research.

The preceding pages, therefore, have provided an introduction to the research. A more detailed statement of its aims, objectives and scope will follow in the research framework chapter, which will then lead to a review of the literature on CNC manufacturing methods for machining asymmetric cylindrical rotational parts with prismatic components, and on interoperability in CNC manufacturing processes. It is to this body of literature that this research seeks to make a contribution. The thesis then explores a novel approach for the implementation of cross-technology interoperability for CNC manufacturing as a means of filling the current gap in interoperability and, to implement this approach, a prototype of a cross-technology CNC interoperability system (XTSys) is developed. Then, in the experimental part of the research, the XTSys prototype is evaluated by machining test components on a CNC turn-milling machining centre as well as a 4-axis CNC vertical machine. Finally, a discussion of the research is presented followed by the research conclusions and avenues for potential future work. Figure 1.3 shows the organisation of the chapters of the thesis.

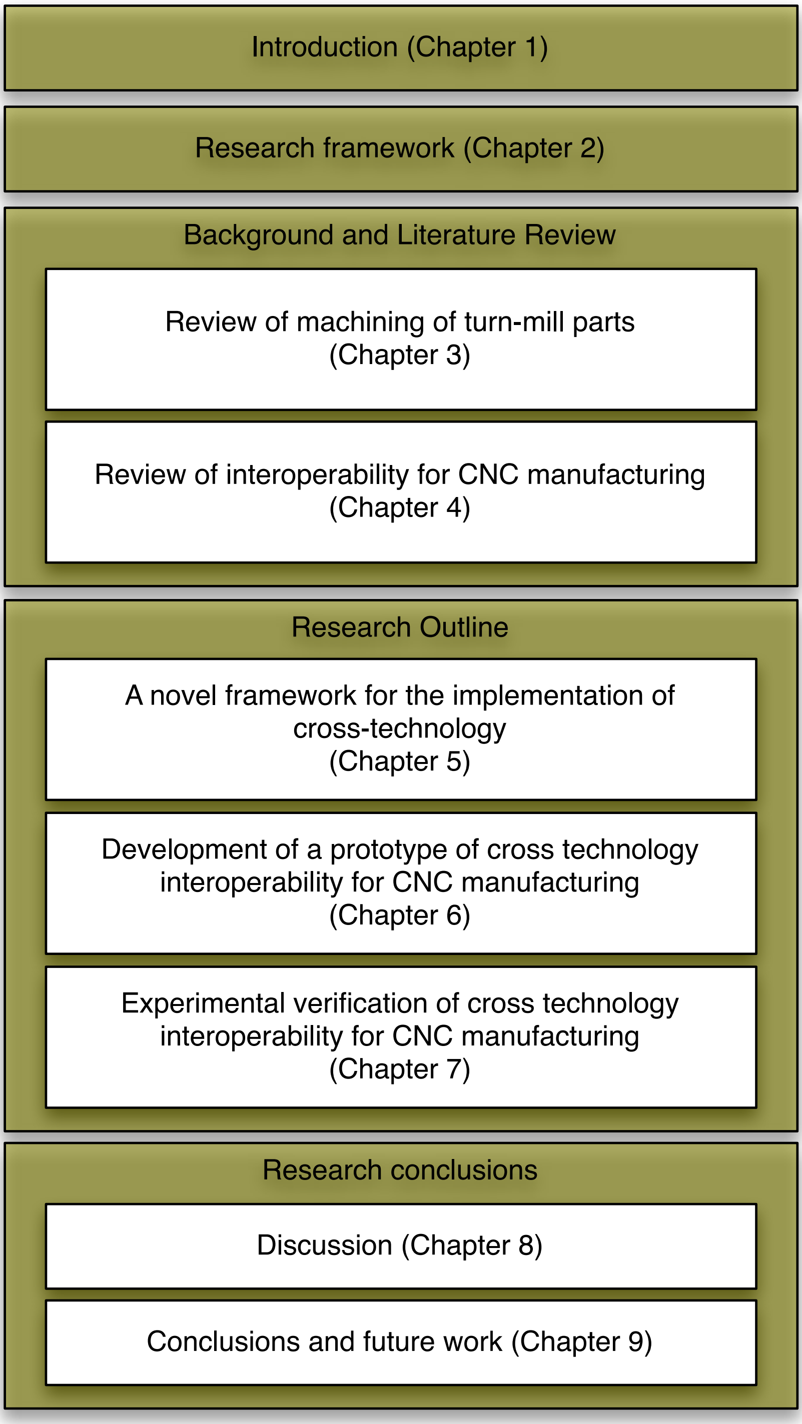


Figure 1.3: Organisation of thesis chapters

2 Research framework

2.1 Introduction

This chapter describes the overall framework of the thesis. The context of the research is first established to highlight the importance of the realisation of the research aim. This aim is then clearly defined, followed by the methodology for reaching the aim. The scope of the research is then presented to express the perspective within which the topic is considered. Finally, the boundaries posed by technical and resource limitations are set out.

2.2 Context of the research

CNC machines with different technologies are often operated as islands of automation that have no links. This causes the agility of a manufacturer to be limited to the context of a single technology (Zhang et al., 2013). Considering that many components can be machined on machines with various technologies; cross technology interoperability would increase the resource pool and thus the agility of manufacturing companies many times over. The following scenarios highlight the advantages of enabling interoperability across CNC machine tools:

- Where there is a requirement for machining legacy parts on new machine tools: these are parts for which the design of the manufacturing process was carried out a long time ago and no information about the decision making process has been logged. For example, legacy aerospace parts which were designed to be used for 20 years but are still required for airplanes in service. In the absence of an information standard for keeping track of manufacturing decisions taken throughout the process, the only data available for these components is the original NC files, prepared for machine tools that may no longer be available. Interoperability would allow the code written for old machines to be executed on new machines without the need to repeat the manufacturing decision making process (Zhang et al., 2012). Cross-technology interoperability would take this flexibility even further as new technologies may supercede and replace older technologies thus making the ability to use new machines, even more critical.

- Machining the same part with different CNC machines on the shop floor: In large CNC manufacturing companies, there exists a variety of different CNC machines, which use different technologies. Most of the research today, has been done to enable transfer of information between machines with the same technology (Schroeder and Hoffmann, 2006, Liu et al., 2007). Such companies would require interoperability between different CNC technologies in order to be able to manufacture a part on machines with various technologies. The economic advantages of the resulting flexibility make interoperability between CNC machines as important as interoperability between CAx systems.

- Reducing the cost of CAD/CAM systems for different CNC machines: Each machine with a specific controller needs a specific postprocessor to generate a code which will work with that machine (Hardwick and Loffredo, 2006). This means there is a large cost associated with providing CAD/CAM systems for different CNC machines. In addition, whenever a new machine or new controllers are installed, then a new post-processor for the CAD/CAM system is needed as well. By having a system that is independent from CAD/CAM systems on the shop floor, it is possible to reduce the overall investment in CAD/CAM systems.

- Using a different machine in a breakdown situation on the shop floor without redesigning the code from the beginning of the CAx chain: In the event of a machine breakdown, production will need to be stopped until either the machine is fixed or another way of continuing production is found. One possibility is to use another CNC machine to manufacture the job but, to do this, new machine code which is compatible with the new machine would be required, and this means reprogramming the code from the beginning. If there is a system that can generate the code directly on the shop floor without any reprogramming from the beginning, it will be possible to save significant time in restarting the production.

2.3 Aim of the research

The aim of this research is to **propose, specify and evaluate a new approach for achieving interoperability between CNC machine tools with different technologies** so that they can produce the same part to the required specification.

2.4 Objectives and methods of the research

A number of objectives have been identified to allow the progress in achieving the aim outlined above to be managed and monitored. The collective realisation of these objectives will equate to achieving the overall aim:

- Identify the specific interoperability gap in CNC manufacturing between CNC machines with different technologies by reviewing state-of-the-art of CNC manufacturing with regards to interoperability and different technologies that would be capable of producing a given set of parts.

- Propose an approach to enable interoperability between CNC machines with different technologies for the production of the same part. This objective can be achieved by:

    - Reading the source machine code and refining the process semantics to discover the information needed to generate new code for the destination machine.

    - Utilising semantic substitution to convert the process logic to one that is suitable for the destination machine.

    - Reformatting the syntax of the translated process logic to match the requirements of the destination machine.

- Validate the framework by writing a prototype adapter to implement the approach that works with two CNC machines with different technologies (milling and turn-mill).

- Verify the prototype by machining a part with a CNC 4-axis machining centre and a CNC turn-mill centre.

2.4.1 Literature

The literature review will survey seminal works and research methodologies in the field to identify gaps in the current knowledgebase so as to help focus and refine the research question. In order to identify the research gap that forms the foundations of the specification requirements for the current research, chapters 3 and 4 present an evaluative review of interoperability and CNC manufacturing methods for asymmetric cylindrical rotational parts with prismatic components as representative parts that can be machined using two different technologies.

2.4.2 A novel approach for implementation of interoperability

Based on the gap in interoperability identified from the literature review, a novel framework is developed in chapter 5 to establish the requirements for interoperability between CNC machines with different technologies. The structure of the Cross-Technology CNC interoperability system (XTSys) is developed to implement this framework.

2.4.3 Development of a prototype implementation of XTSys

A prototype implementation of the XTSys is then generated to validate the proposed framework. In chapter 6, the interface and code of XTSys are developed to implement the framework.

2.4.4 Experimental verification of XTSys

In order to verify the prototype for interoperability developed in this research, in chapter 7 experimental parts are chosen to be machined with different CNC machines using codes generated through XTSys.

2.5 Scope and Boundaries of the research

The perspective within which the research method is realised can be defined using the following scope:

- The focus will be on data transfer allowing interoperability between machines. To use the terminology standardised as the Open System Interconnection (OSI) network model (see Figure 2.1 for a list of the layers), it is assumed that the physical connection between the machines and low level software layers on this link already exist (ISO/IEC 7498-1, 1994).

- Determining the machinability of the programs on a machine tool is, in itself, an interesting research problem. In the scope of this work, it is assumed that the parts for which the programs are transferred are machineable on both the source and destination machines. Proving machinability, will thus not be considered in this thesis.

- The data transfer between different technologies in this research will focus on transferring STEP-NC code from one machine to another. Old machining code will not be considered in this research.

- In this research transferring data between CAD/CAM systems will not be considered because the proposed system will be used on the shop floor and will work without the use of CAD/CAM systems.

In order to keep the research manageable, a number of boundaries have been identified. These boundaries with respect to CNC machine tools, manufacturing standards and CAPP/CAM are shown in Figure 2.2. These boundaries are justified as follows:

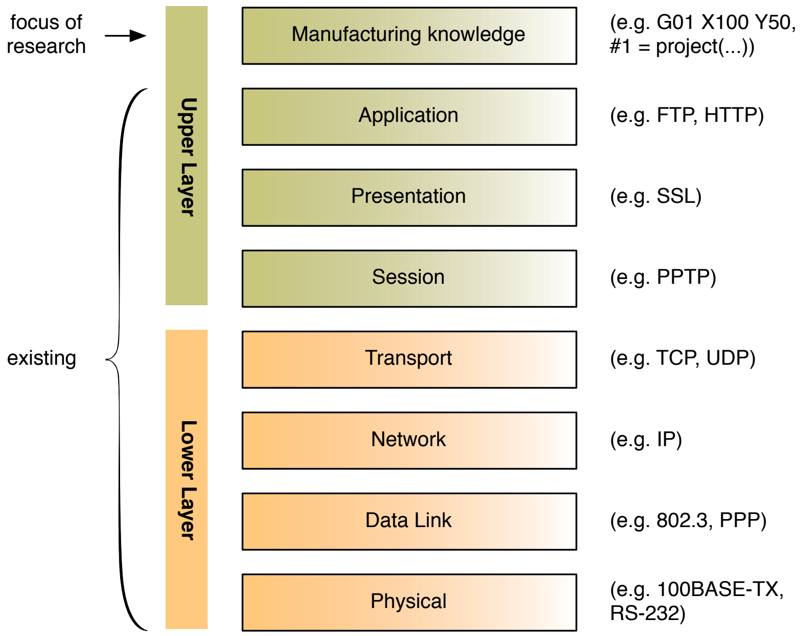


Figure 2.1: OSI layer model adapted form (ISO/IEC 7498-1, 1994)

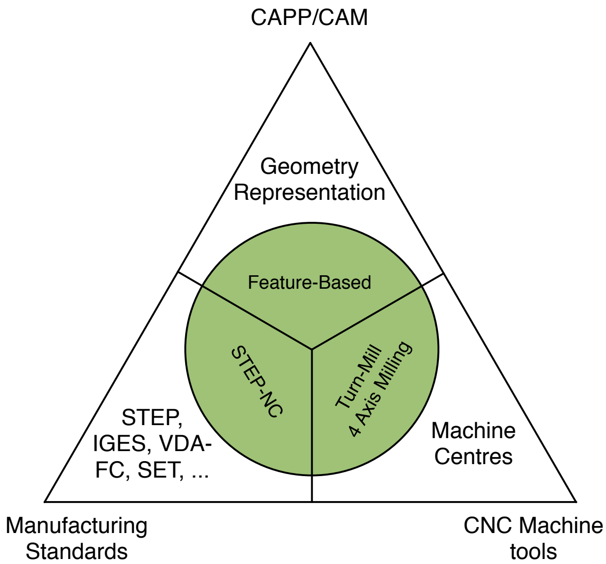


Figure 2.2: Research boundaries

- **CNC turn-mill machine and CNC 4-axis vertical machining centre**

A variety of CNC machine tools with different capabilities and different controllers are involved in the production of complex parts. To illustrate the interoperability system for this research, a CNC turn-mill machine and a 4-axis CNC vertical machining centre have been chosen as two CNC machine tools that can manufacture similar parts with different cutting tools and processes.

These two technologies were chosen because they are the most common machines that are used in industry and because there is a mature standards framework for these two technologies. In addition, three factors make realisation of interoperability between these specific technologies relevant:

i. Modern turning centres have gained milling capability through the introduction of live tools and additional linear axes of movement. This allows parts that used to require multiple setups on lathes and milling machines to be manufactured in a single setup. Therefore the capability of translating the old milling programmes for combining with the turning portion of multiple setup components can be used to use the advantages of these modern machines without requiring extensive labour costs.

ii. In milling components that have many cylindrical features, the use of a turn-mill machine, instead of the original milling machine for which the programme was written could bring about speed benefits. This is due to the fact that machining round shapes on the turning machine is generally faster.

iii. The tolerances achievable in milling machines have become better over the recent years and as such, it is possible to machine parts with few round features on a milling machine without sacrificing precision. In many cases, the quality of the finished part has no discernable difference with a similar part produced on a turning machine.

It is also noteworthy that the approach proposed can be adapted to work with other machines with different technologies by incorporating their syntactic and semantic profiles within the system. This process is conceptually identical to what is shown for 4-axis milling and turn-mill processes.

- **STEP-NC in interoperability standards**

A large number of standards have been put forward in CNC manufacturing to regulate information formats and its supply to CNC machine tools (Nassehi, 2007). STEP-NC has been chosen in this research as the means with which information from CAPP and the machining process are sent to a CNC machine.

The STEP-NC standard was chosen because it provides valuable data structures to store manufacturing information like features, operation and machine information, and this facility is important for the design of a framework for implementing interoperability based on semantic substitution and the development of a prototype of the framework which will work with both selected machines. The work could be extended to cover other standards as well but this would require additional research.

- **Feature-based technology in CAPP/CAM systems**

Feature-based CAx has been a major area of research for over 30 years, and it is one of the central topics in CAD/CAM and CAPP integration (Xu et al., 2011). Features represent the highest level of information maintained in computer systems about part geometry and have been used for many purposes in the CAD/CAM/CNC chain, as an example, the use of features in costing (Saravi et al., 2008) shows their flexibility. Hence this approach was chosen in this research because almost all CAPP systems function on the basis of features which are mostly the same whatever the technology of the CNC machine actually used for the machining.

3 Review of machining of asymmetric rotational parts

3.1 Introduction

This chapter sets out a definition of processes for manufacturing cylindrical rotational parts with prismatic components and provides an overview of CNC machine tools and their technologies to identify the domain within which interoperability is pursued in this research.

3.2 Manufacturing processes

The term "process" can in general be defined as a change in the properties of an object, including geometry, hardness, state, and so on. The general manufacturing process model is illustrated in Figure 3.1.

Material flow can be divided into three main types:

- Through flow, (shown in Figure 3.2) corresponding to conserving processes: Processes in which the mass of the input material and the output product are the same. Deep drawing, casting and forging are examples of this flow.

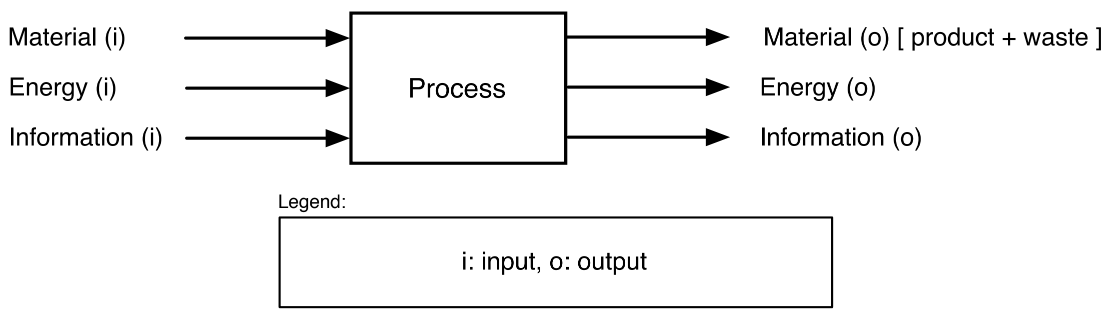


Figure 3.1: The general process model

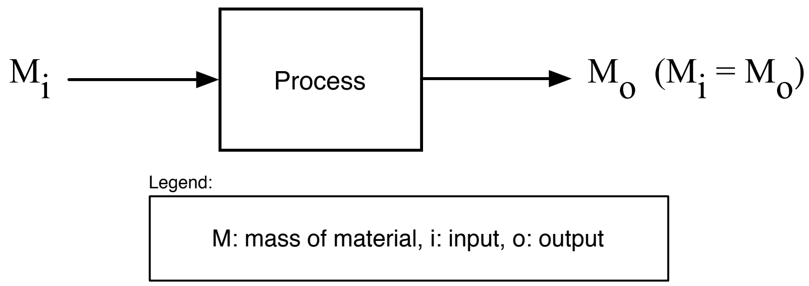


Figure 3.2: Through flow.

- Diverging flow, where the mass of input material is equal to the mass of the product plus waste (see Figure 3.3). Machining, punching and grinding are examples of this flow.

- Converging flow, where the mass of the product is equal to the sum of more than one input material (see Figure 3.4). Welding, rapid prototyping and bonding are examples of this flow.

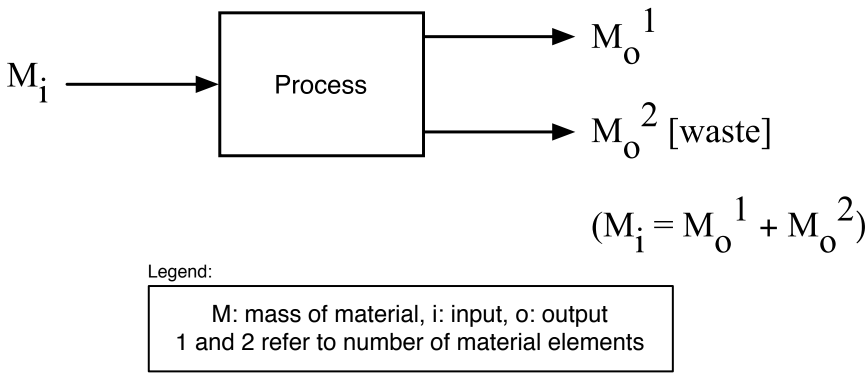


Figure 3.3: Diverging flow.

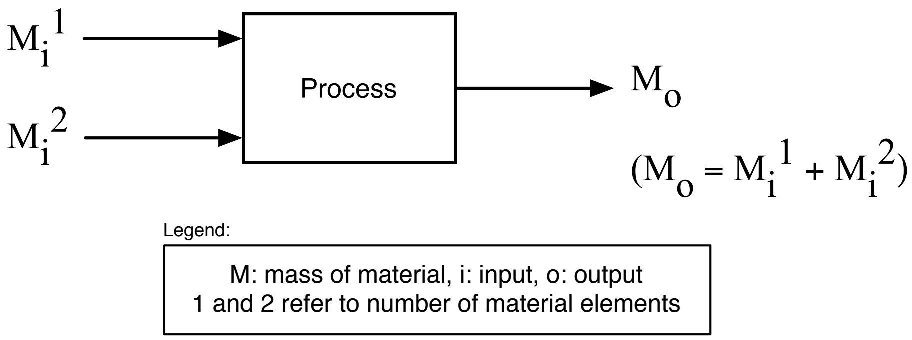


Figure 3.4: Converging flow.

A comprehensive classification of manufacturing processes was done by Kalpakjian and Schmid (Kalpakjian, 2014). They classify processes into six subsections with casting, bulk deformation, sheet metal, polymer, machining & finishing and joining. Figure 3.5 illustrates the classification of manufacturing processes together with their flow type. Processes with a diverging flow (and sometimes those with a converging flow), often work based on controlled motion of a tool against the workpiece. In CNC machines this motion is controlled by a computer and thus for such technologies correct programmes are essential to achieve production goals. It is therefore in such technologies where interoperability is crucial.

Within these technologies, machining is especially important as most products are either machined or produced by tools or machines, components of which were machined. As Holland (1989) indicated: "Machine tools are in fact the 'mother' or 'master' machines, the machines that make all machines. Every manufactured product is made either by a machine tool or by a machine that was made by a machine tool" (Holland, 1989). Of the types of products that are machined, 47 per cent are rotational and 75 per cent of rotational parts have some additional prismatic features (Rosso et al., 2004, Tavakoli, 1993). An overview of the technologies that are required to produce such components follows.

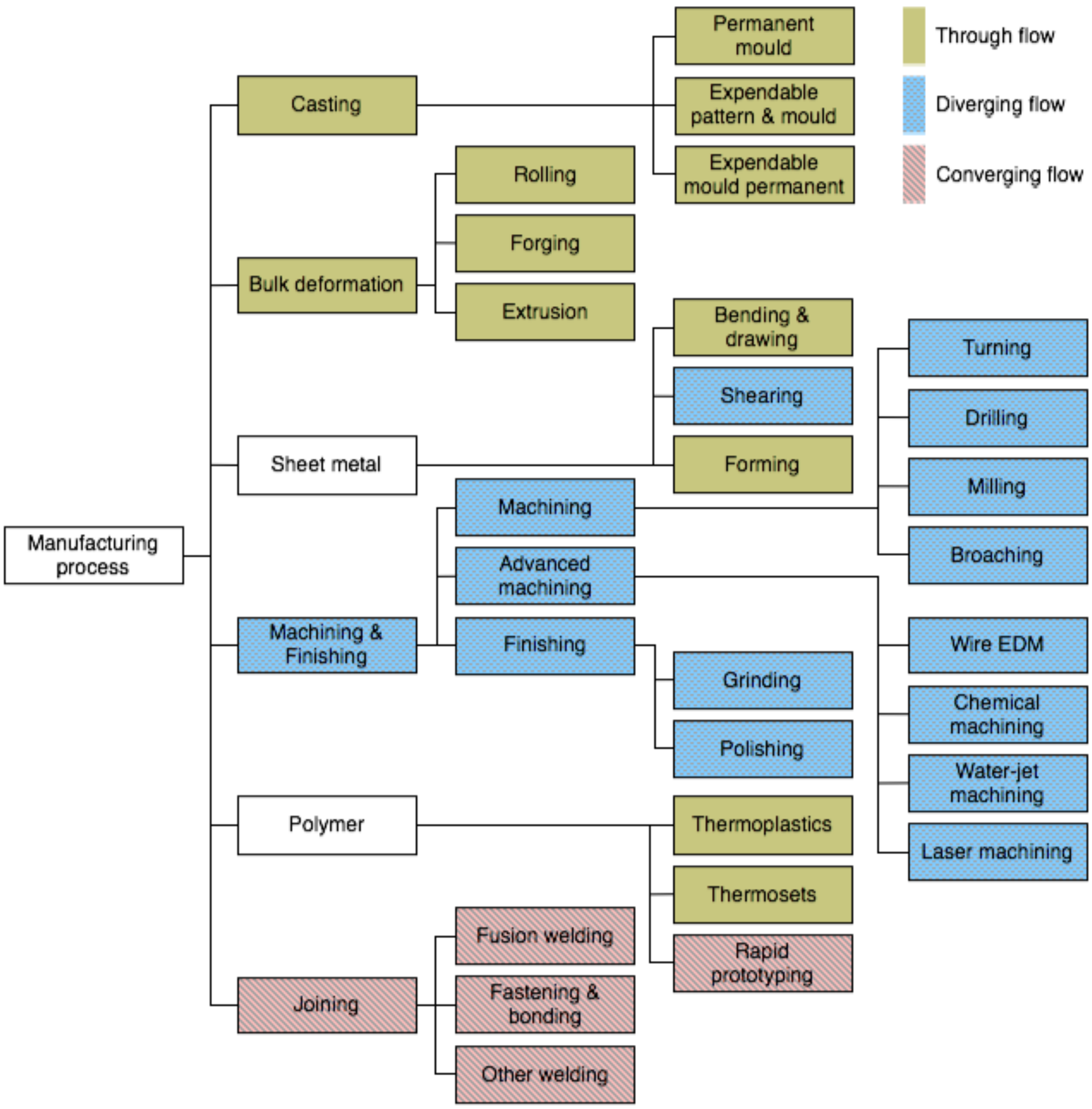


Figure 3.5: Manufacturing processes classification adapted from (Kalpakjian, 2014)

3.2.1 Turning process

Turning is the removal of material from the outer, or in some cases inner, diameter of rotational cylindrical parts where the cutting tool that is typically a non-rotary tool, removes material from workpiece by moving more or less linearly while the workpiece rotates. A typical turning machine is comprised of a spindle that holds and turns the workpiece, a tool holder, a machine bed and mechanical or electrical motors to allow the rotating parts to revolve and linear moves to be made. Figure 3.6 illustrates the typical components of a manual lathe. The first metal lathe was invented by John Wilkinson in 1775 for boring cannons and it was later adapted to be driven by a steam engine during the industrial revolution (Stephenson and Agapiou, 2006).

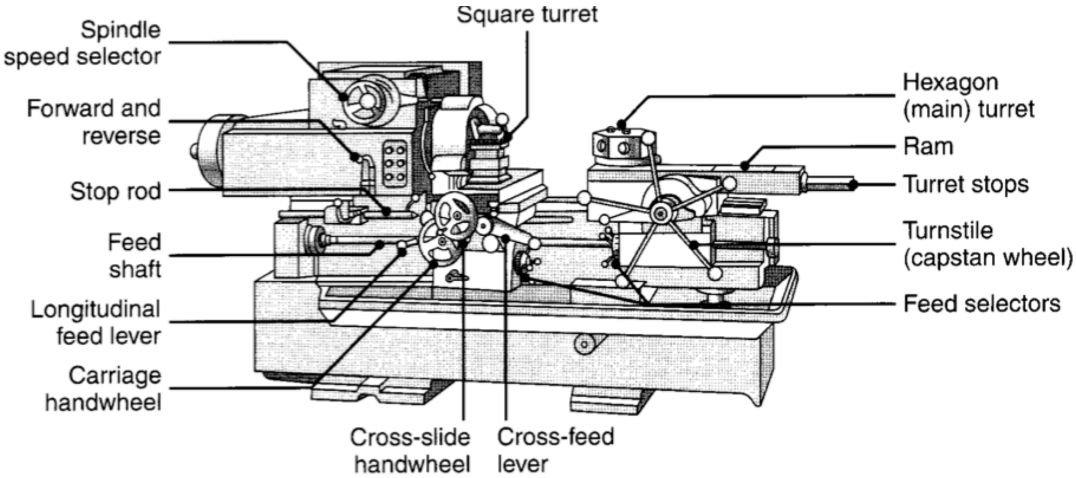


Figure 3.6: Manual lathe adapted from (Kalpakjian, 2014)

Various turning operations are carried out by turning machines. The five principal types of operations are facing, grooving, contouring, threading and knurling (ISO 14649-12) as shown in Figure 3.7. Every part produced using turning technology is manufactured using a sequence of these operations.

A manufacturing system capable of supporting these individual types of operation would thus be capable of supporting turning technology in its entirety.

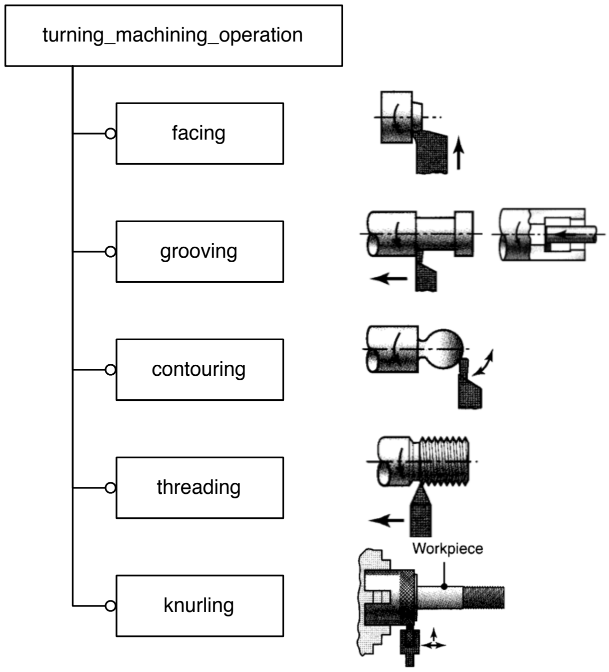


Figure 3.7: Turning machining operation

3.2.2 Milling process

The second most prominent machining technology is milling. Milling is a material removal process where a rotary cutter removes material from a fixed workpiece through feeding the workpiece to the rotary cutting tool (Benhabib, 2003). The process is typically used for nonrotational objects. A typical milling machine contains a worktable which holds the workpiece, a spindle that rotates tools, a machine column and mechanical or electrical motors to allow the axes to move. Figure 3.8 illustrates the typical components of a manual milling machine.

Eli Whitney invented the first metal milling machine in 1818. Since Whitney was among a group of contemporaries all developing milling machines at about the same time from 1814 to 1818, it has been argued that no individual can properly be credited as being the sole inventor of the milling machine (Meyers and Slattery, 2001).

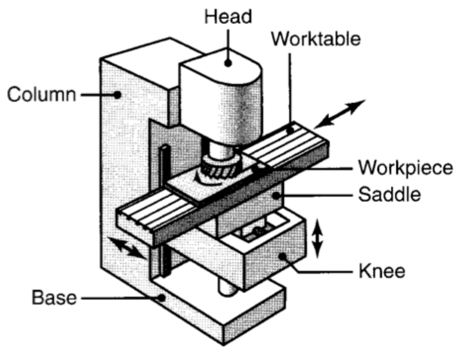


Figure 3.8: Manual vertical milling machine adapted from (Kalpakjian, 2014)

According to ISO 14649 part 11, the three principal types of milling operations are plane_milling, side_milling and bottom_and_side_milling as shown in Figure 3.9. Parts produced using milling technology are manufactured using a sequence of these three operations. A manufacturing system capable of supporting these individual types of operation would thus be capable of supporting milling technology in total.

3.2.3 Drilling process

In addition to milling and turning, drilling type operations complete the most used of machining processes. These types of operations create round holes in a workpiece. Drilling type operations can be carried out either by a rotating tool attached to a spindle that feeds to the workpiece, or by fixed tool against which a rotating part is fed (Benhabib, 2003).

According to ISO 14649 part 11, there are five different drilling type operations: as drilling, boring, back boring, tapping and threading. In this research basic drilling will be considered due to lack of tools for experiments on other types of these operations.

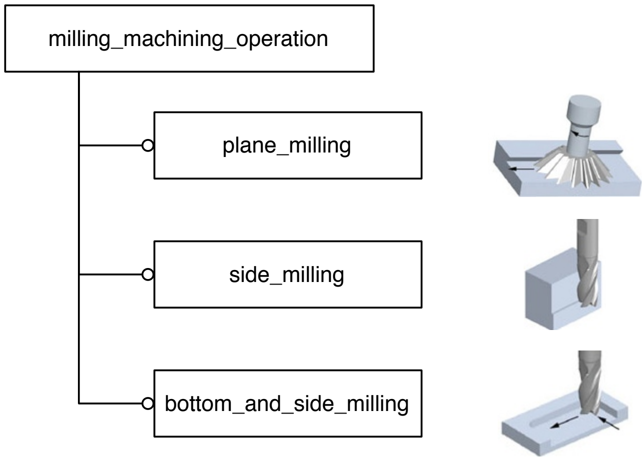


Figure 3.9: Milling machining operations

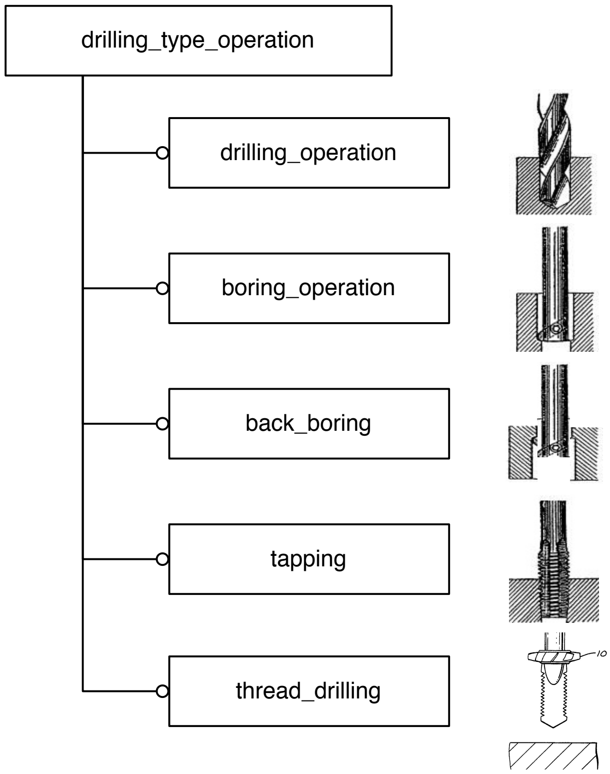


Figure 3.10: Drilling operations

3.3 CNC machine tools

A machine tool is a powered mechanical device, typically used to produce metal components by machining the selective removal of metal. Automating the control of the tool motion in machining allow parts to be made in a repeatable manner.

John T. Parsons and Frank L. Stulen pioneered the first numerical control (NC) system for machine tools in the 1940s. They were the first to use a cardpunch computer to control the movements of the axes on a milling machine (Kochan, 1986). MIT and IBM worked together in inventing the first practical NC machine in 1952 (Dorf and Kusiak, 1994). John T. Parsons was granted a patent for his work in using numerically controlled motors to direct the machine tool positioning in 1958 (Parsons and Stulen, 1958).

CNC machines were developed in the 1970s with the introduction of minicomputers and CAD drawing software to support the development of on-machine programmes to enable machining of different parts (Talavage and Hannam, 1988). This allowed parametric programming of the machines as well as the bringing about the ability to reprogramme machines very quickly.

CNC machines can be classified in a number of different ways (Vichare, 2009); the main attributes based on which the machines can be categorised include:

i. Machining process

CNC machines can be classified based on their machining process such as turning, milling, turn milling, drilling, inspection, punching, laser, welding, tube bending, grinding, etc. It should be recognised that a number of these machining processes are now combined in the form of hybrid manufacturing workstations (see Figure 3.11). The machining processes considered in this research are turning, milling and drilling.

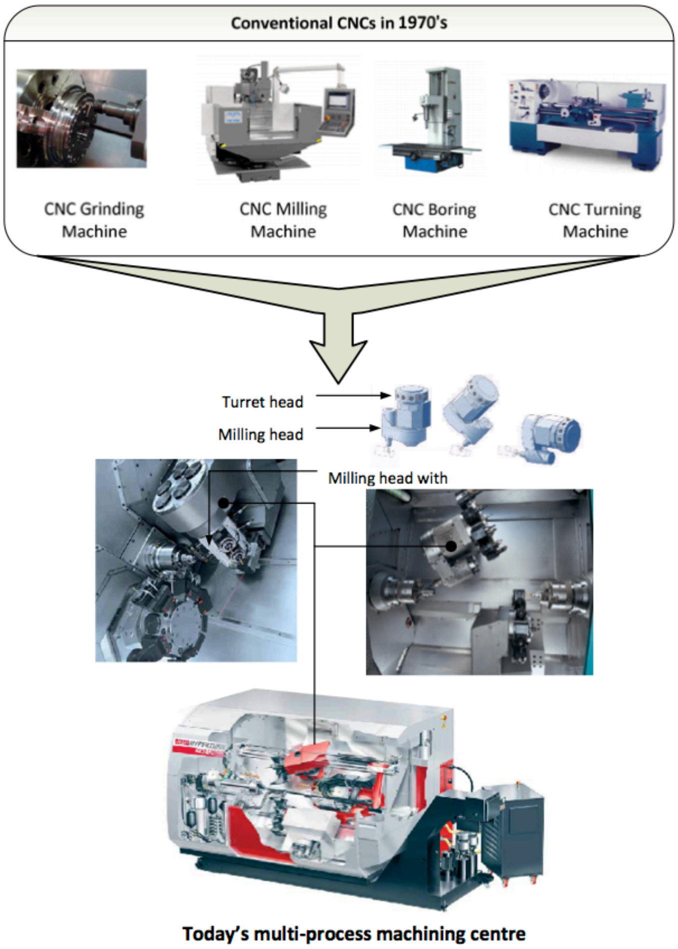


Figure 3.11: Convergence of manufacturing processes on a single platform (Emco.Maier.Ges.M.B.H, 2007)

ii. Number of axes

An important property of machine tools is their number of axes of movement. The conventions for linear axes are X, Y, Z and for rotary axes A, B, and C. In addition, machines can be designed with more than six axes, and have other letters such as U, V and W to describe their axis of motion. Figure 3.12 illustrated the typical linear and rotary axes used in machine tools.

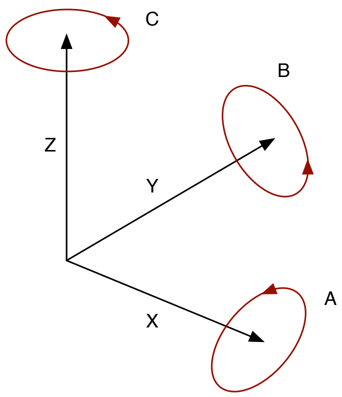


Figure 3.12: Typical Linear and Rotary Axes on Machine Tools.

Most CNC machines work with multiple axis movements in order to achieve a wide range of positions and orientations of the tool in respect to the machined part. One method of CNC machine classification, therefore, is through the number of axes that can be used to control the tool and workpiece interaction.

For example the 5-axis vertical machining centre in Figure 3.13(a) has two linear axes of movement on the table (Labelled X and Y axes), a linear axis of movement on the spindle (labelled Z axis) and two rotary axes of the movement on the table (A and C axes) which allow 5 degrees of freedom on the tool-workpiece interaction position and orientation.

The turn-mill centre in Figure 3.13(b) has three linear axes of movement on each turret head (X1, X2, Y1, Y2, Z1, Z2), a linear axes of movement on the counter spindle (Z3), rotational axes on both spindles (C1, C2) and rotational axes on each turret head allowing 6 degrees of freedom tool access to the component.

The equipment that is used in this research includes a milling machining centre with 3, 4 or 5 axes of movement and a turn-mill machining centre with 4 axes of movement.

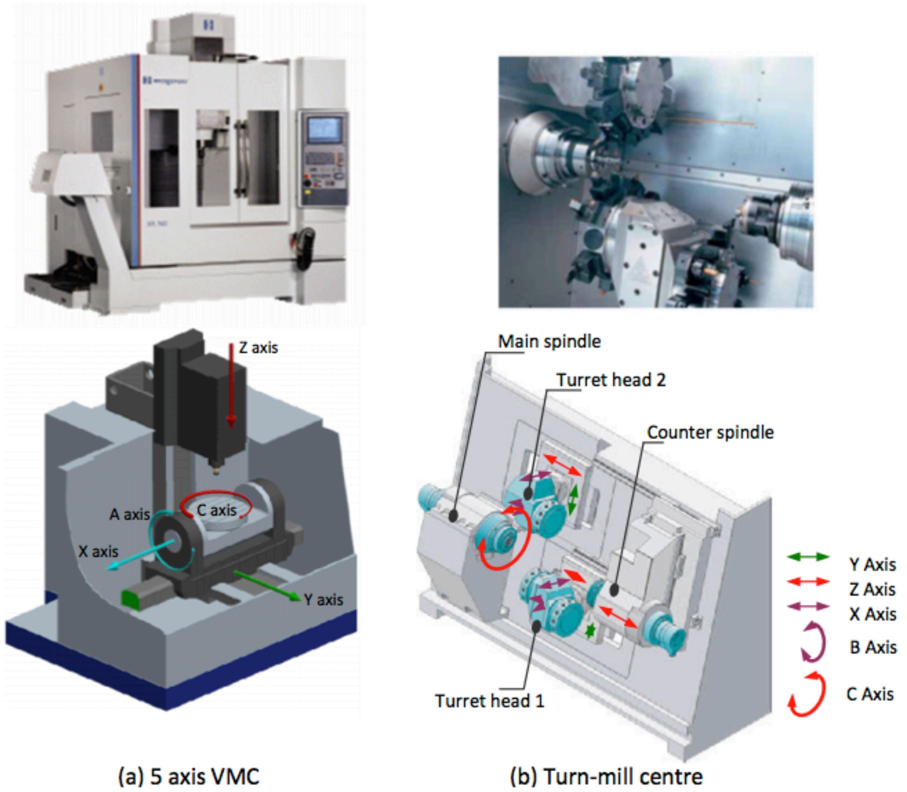


Figure 3.13: Multi-axis machining centres
adapted from (INDEX, 2013, Hardinge, 2013)

iii. Spindle arrangement

A CNC machine can be classified according to whether the spindle is arranged in a vertical or horizontal direction (Stenerson and Curran, 2005). In a vertical spindle orientation, the tools are held in a vertical position so that as the table upon which the part is positioned moves the part will be machined in the horizontal plane. Alternatively, where the spindle is arranged horizontally, machines are generally built with a universal table which is able to effect a rotational movement for machining different aspects of the part. Figure 3.14 illustrates vertical and horizontal machining centres. 3 to 5 axis milling centres that are used in this research are vertical the 4-axis turn-mill centre has a horizontal spindle.

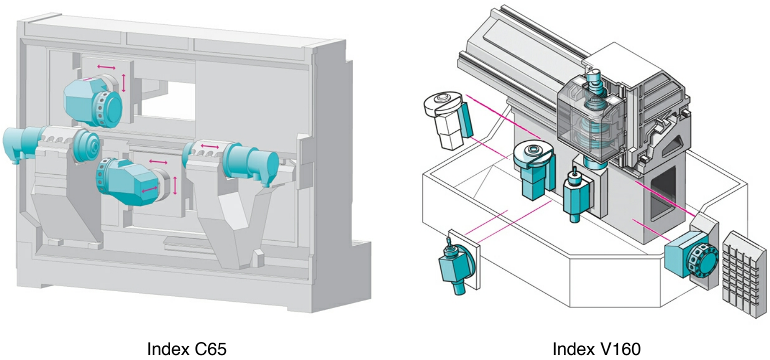


Figure 3.14: Classification according to spindle arrangement (INDEX, 2013)

iv. Number of spindles

The number of spindles is another method for CNC machine classification. Multi-spindle CNC machines are designed to enable a more rapid production process by enabling the machining of multiple parts on a single machine visit. Figure 3.15 shows an example of a two spindle turning centre (Index C65), and an Index MS32, with multiple spindles. While multi spindle machines have more machining capability, in performing individual manufacturing operations they are identical to single spindle machines, therefore, single spindle machining centres were chosen for this research.

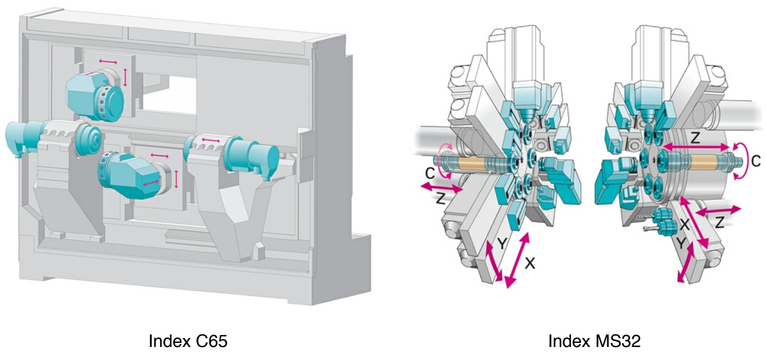
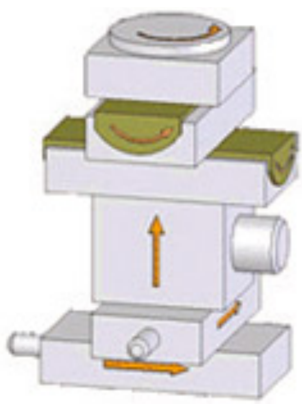


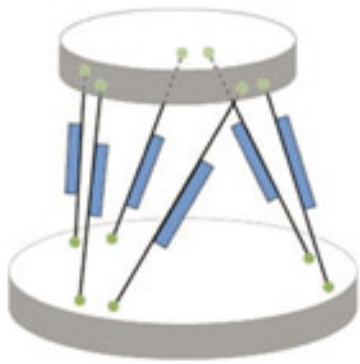
Figure 3.15: Classification according to the number of spindles (INDEX, 2013)

v. Kinematic configuration

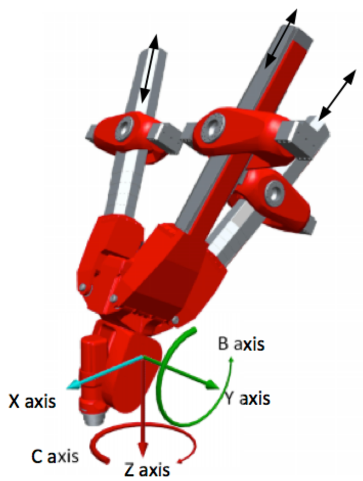
Kinematics is the study of the motion of bodies (objects) and systems (groups of objects), while ignoring the forces that cause the motion. This is another classification for CNC machines that can be used for spindle and table movement (Vichare, 2009). Figure 3.16 illustrates three different types of kinematic configuration. As the focus of this research is on milling and turning technology, the serial kinematic configuration is relevant as the typical configuration of these types of machines.



Serial kinematic configuration



Parallel kinematic configuration



Hybrid configuration

Figure 3.16: Kinematic classification (Exechon, 2013)

3.4 Overview of CNC Milling machining centres

Machining centres can be categorised into eight different types, from two-dimensional to hexapod type machining centres. These types are depicted in Figure 3.17. The machine tools applicable to this research are described in the following sections below namely 3-axis, 4-axis and 5-axis machining centres.

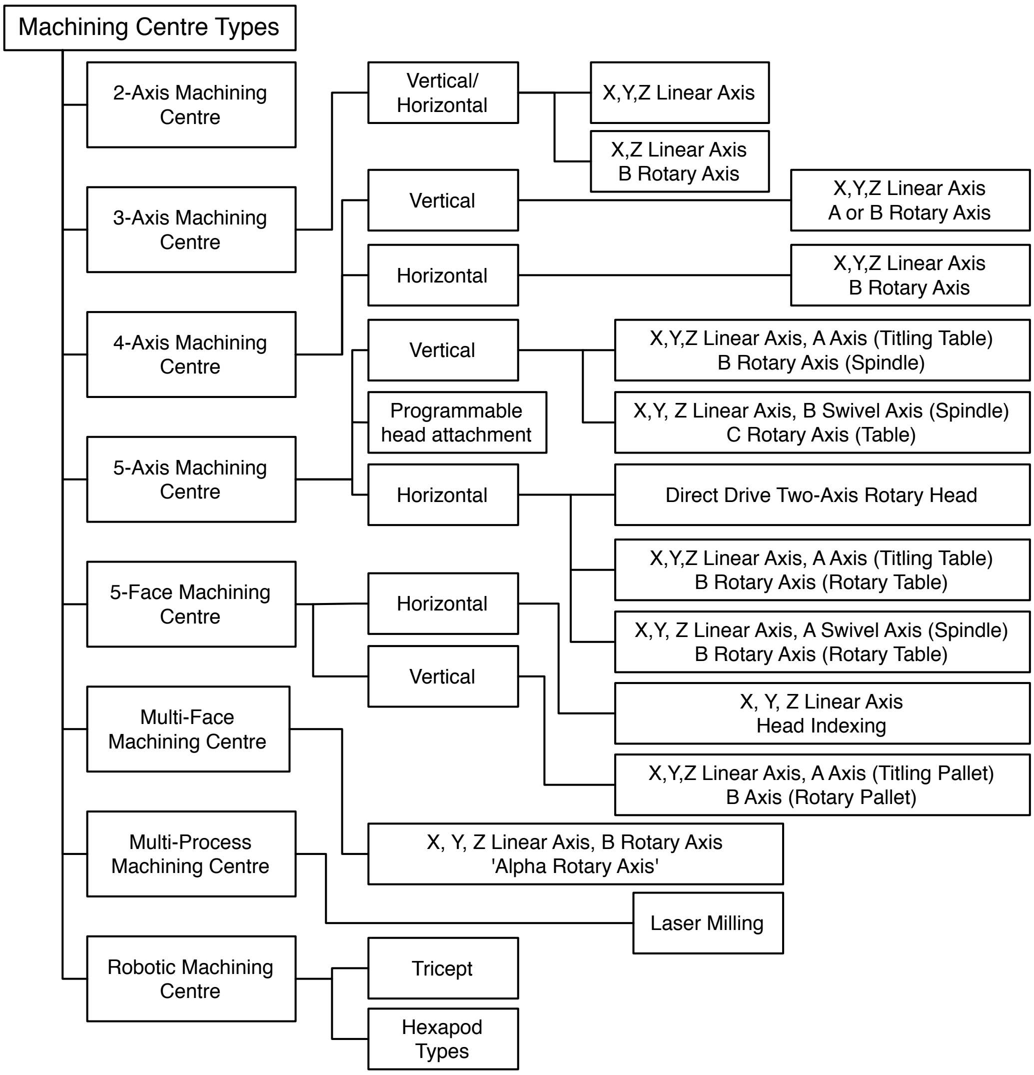


Figure 3.17: Machining centre types according to number of axis

3.4.1 Three axis machining centres

These machining centres allow the tool and workpiece interaction to be moved in three perpendicular axes of movement, allowing tool access to a cuboid machining on the machine table. The simple single spindle, single table vertical machining centre is among the most common CNC machine tools in the industry. An example is shown in Figure 3.18.

Multi-spindle (see Figure 3.19a for a twin-spindle) and multi-table (see Figure 3.19b for a multi-table) variants have been developed to increase productivity. The axes configuration in milling machines is defined such that the positive Z direction points into the spindle of the machine in right hand coordinate system (As shown in Figure 3.20).

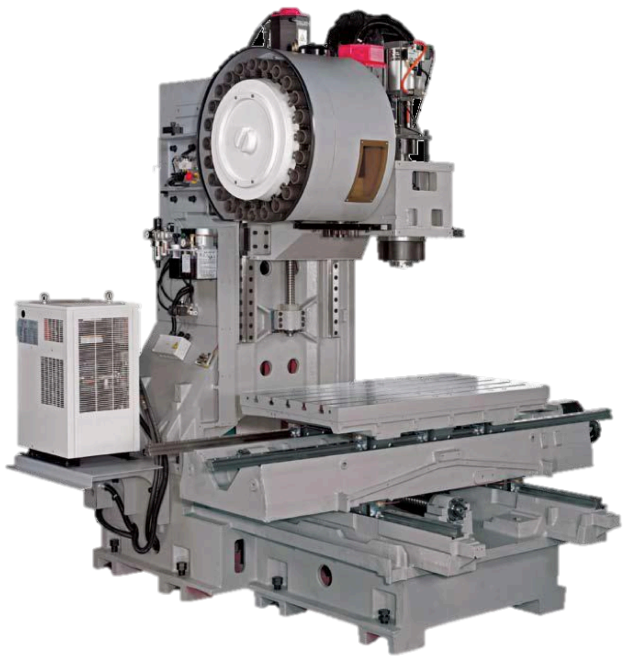


Figure 3.18: Single spindle verticals machine with tool magazine
(KENTCNC, 2013, HAAS, 2013)

a) Vertical machine centre with twin spindle

b) Vertical machine centre with twin table

Figure 3.19: Machining centres with twin spindle and twin table (QUASER, 2013, STAMA, 2013)

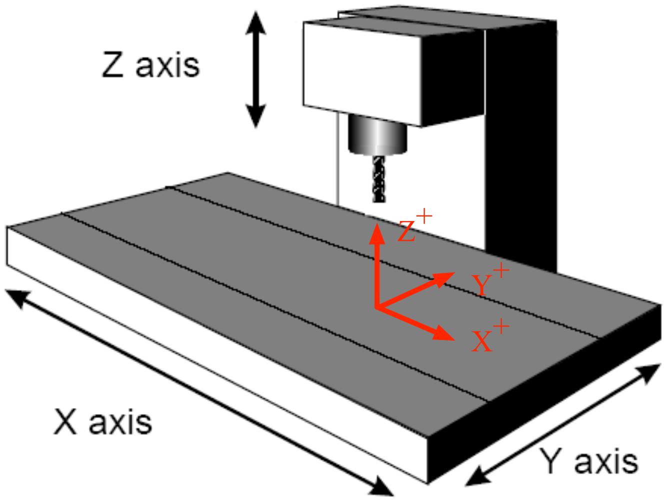


Figure 3.20: The axes configuration in milling machine

3.4.2 Four axis machining centres

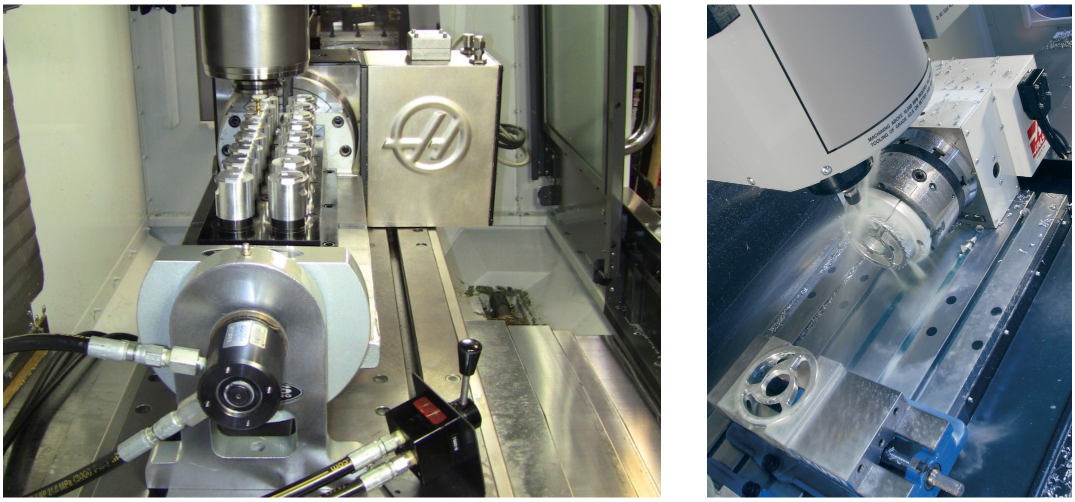
Four axis machining centres are normally built by adding a rotary axis to the 3-axis milling machine. An example of this type of machine where the rotary axis has been added after the construction of the machine is shown in Figure 3.21.

This feature makes one set up machining possible for some work pieces that require multiple setups on three axis machines.

3.4.3 Five axis machining centre

Adding a second non-parallel rotary axis to a four axis machining centre results in the emergence of the five axis machining centre.

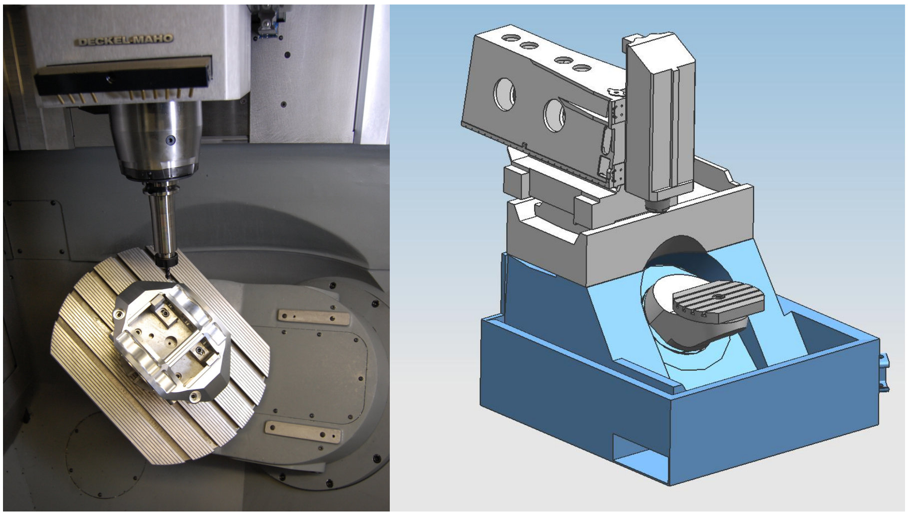
Some common configurations for adding the two rotary axes are rotary table and swivelling spindles (see Figure 3.22 and Figure 3.23 for examples).



a) 4-axis machine centre with trunnion table

b) 4-axis machine centre with rotary table

Figure 3.21: 4-axis machining centres (Branham, 2009, HAAS, 2013)



Deckel Maho DMU eVo vertical machine centre

Figure 3.22: Five axis machining centre with two rotary axes on the table (DMG, 2013)

Mazak VTC 800/30SR

Figure 3.23: Five axis machining centre with one rotary axis on the table and one rotary axis on the swivelling spindle (Mazak, 2013)

3.5 Overview of Turn-Mill machining centres

In parallel with developments in CNC milling, CNC turning also had major developments in the 1970s with the improvement in machine capability for machining curved contours and improving versatility in the area of cutting tool control.

In conventional turning, it is very difficult to maintain optimum cutting conditions, whereas, in CNC, it is possible to vary the spindle speed with the tool position to maintain economic cutting conditions. Lathes and turn-mill machines are constructed so that the spindle is linked to the workpiece and thus turning operations can be performed. The axis configuration on lathes and turning centres is such that the positive Z direction is pointing out of the spindle (see Figure 3.24). CNC lathes allow the tool, workpiece interface to be moved in two perpendicular axes of movement. One of the axes, Z, is parallel to the axis of rotary part revolution.

CNC lathes are normally classified as vertical or horizontal. Vertical lathes are relatively uncommon and are used for large and unwieldy workpieces (for an example see Figure 3.25).

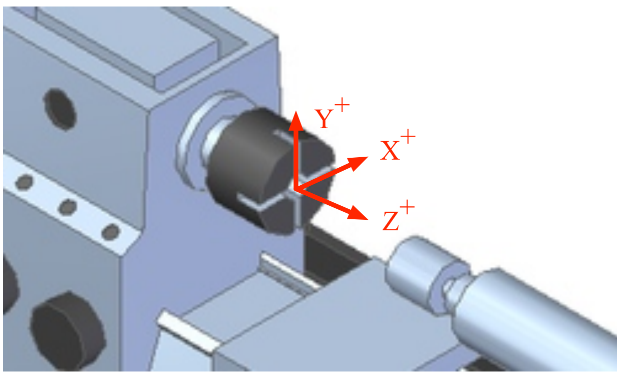
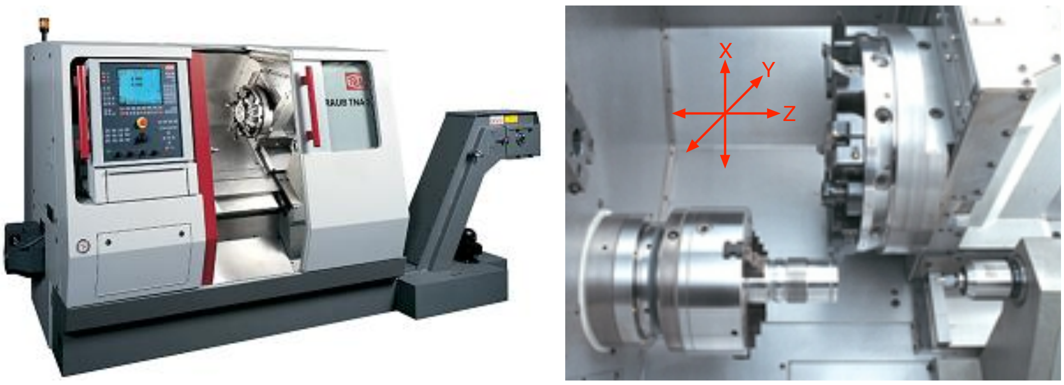


Figure 3.24: Three axes configuration in lathe and turn-mill machine

HNK VTC 50/60

Figure 3.25: An example of vertical lathe (HNK, 2013)

The more convention horizontal lathes have gone through many design changes over the years with slant bed design now being popular (see Figure 3.26 for an example). In this design, chips fall in a pan at the bottom of the machine or on a conveyor to be removed automatically from the machining area.



Traub TNA300

Figure 3.26: 3-Axis turning machine (INDEX, 2013)

A turn-mill centre is a lathe with additional rotary or linear axes of movement and powered tool holders. A turn-mill machine can thus perform both turning operations (where the part is rotated for machining) and milling operations (where the tool is rotated). Figure 3.27 illustrates the implementation of milling operations on turn-mill machines.

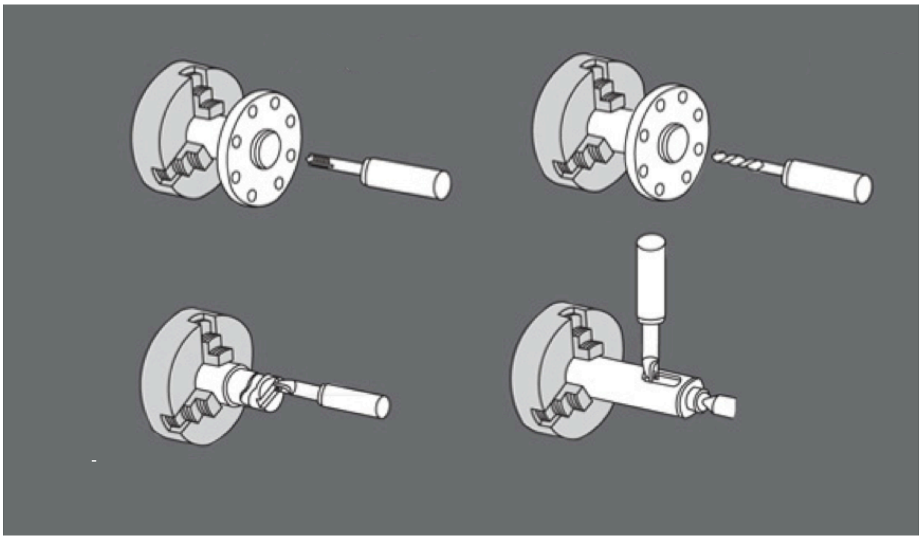


Figure 3.27: Milling operations on turn-mill machines (CubicMachinery, 2013)

Since the late 1980s turn-mill machines have had rotary control of the workpiece (C-axis control); this allows the machine to manufacture slots, cam grooves and helixes. Figure 3.28 illustrates these types of operation.

Figure 3.28: Capability of controlling C-axis (Mas.Tech., 2013)

Turning Machining centres can be categorised according to number of axes, from two to eight. Figure 3.29 illustrates turn and turn-mill machining centre classifications (in which turn-mill technology is highlighted). An example for multiple spindles is shown in Figure 3.30a and a more complex example with multiple spindles and multiple turrets in shown in Figure 3.30b.

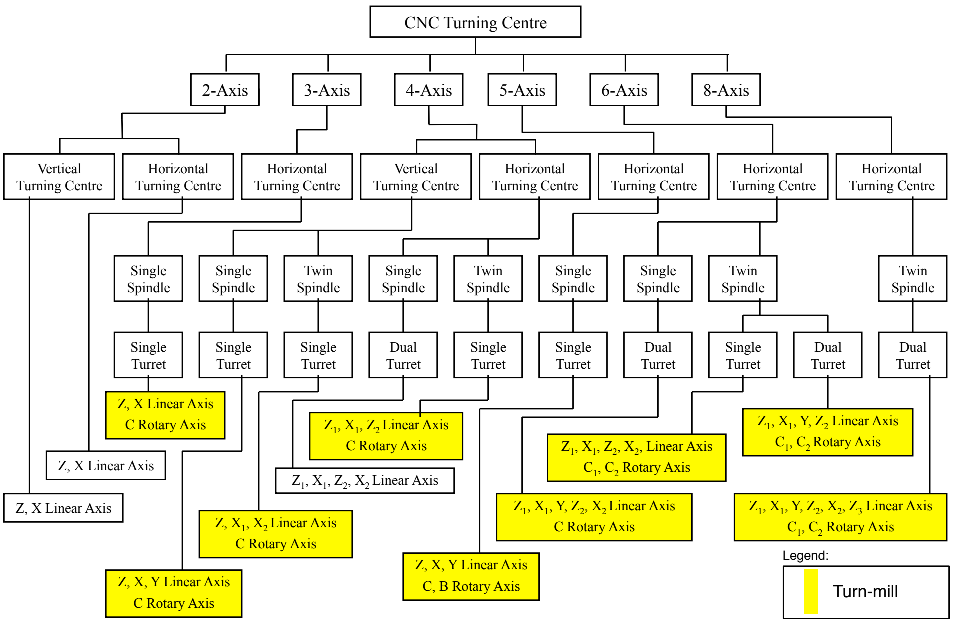
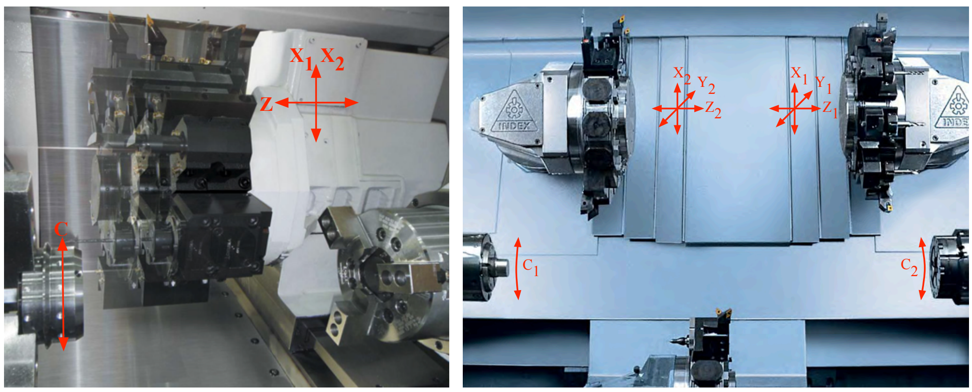


Figure 3.29: Turn and turn-mill machining centre classification



a) 4-Axis vertical turn-mill centre with two spindle

b) 8-Axis turn-mill center

Figure 3.30: Multi spindle and turret turn-mill centre (INDEX, 2013, Okuma, 2013)

3.6 Critical assessment of the latest methods for manufacturing asymmetric rotational parts

Waiyagan and Bohez defined an asymmetric rotational part as “a class of parts having primitive shapes with one common centreline like cylindrical, cone or round shapes including other symmetric and/or asymmetric rotational machining features.” (Waiyagan and Bohez, 2009); Prismatic components are those where machined faces are either parallel or perpendicular to the base of the component. An asymmetric rotational part with prismatic components thus combines both types of features in the same part. Figure 3.31 shows symmetric and asymmetric parts with prismatic components.

While these parts can be machined in multiple setups on multiple machines, some specific machine types have been developed to address the requirements of these parts. In particular, turn-mill centres have been developed to support multifunction machining, with the capability of turning, milling and drilling operations combined on a single machine; with multiple rotary and linear axes of movement controlled at high speed with high accuracy.

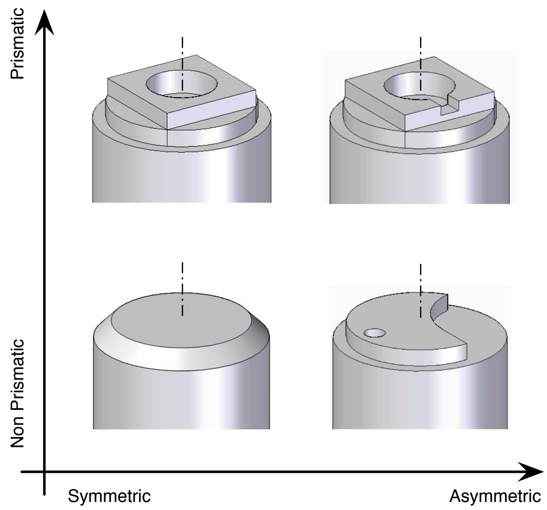


Figure 3.31: Asymmetric rotational part with prismatic component

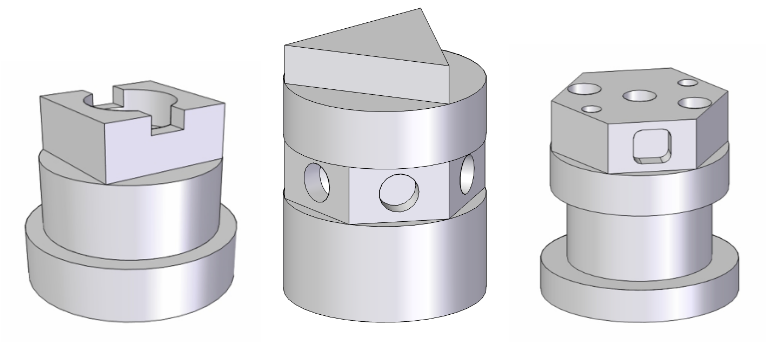
By adding advanced tool magazines and intelligent tool changing to speed up manufacturing with automatic programing for each setup, manufacturing of complex parts is possible with minimal manual intervention. According to chosen technologies and machines in this research, three classes of parts with different features have been chosen to evaluate the applicability of the interoperability between milling and turn-mill technology:

a) Asymmetric rotational parts with prismatic features with a single tool access direction parallel to the axis of revolution;

b) Asymmetric rotational parts with prismatic feature with a single tool access direction perpendicular to the axis of revolution;

c) Asymmetric rotational parts with prismatic feature with multiple tool access directions.

Class (a) can be manufactured by a 3-axis vertical milling centre in a single setup, class (b) can be manufactured with a 4-axis vertical milling centre in single setup and class (c) requires a 5-axis milling centre to be manufactured in a single setup. All three classes can be machined with a turn-mill centre. Figure 3.32 illustrates examples for these three part classes.



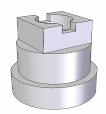
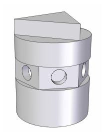
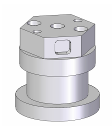
(a) Test part 1 (b) Test part 2 (c) Test part 3

Figure 3.32: Test parts

Table 3.1 shows the number of setups for each part class depending on the machine type and the available axes on each type. As mentioned in section 3.2, a manufacturing system capable of handling individual operations would be able to support combinations of the operations as well.

Thus, by addressing the interoperability requirements of the three defined part classes, cross technology interoperability between milling and turn-mill will be realised. Test parts 1, 2 and 3 which are used to validate the research in chapter 7 represent these three classes.

Table 3.1: Number of setups for each part class according to machine types

Part	**3-Axis Milling Centre**	**4-Axis Milling Centre**	**5-Axis Milling Centre**	**Turn-mill Machine**
	Single Setup	Single Setup	Single Setup	Single Setup
	-	Single Setup	Single Setup	Single Setup
	-	Multi Setup	Single Setup	Single Setup

4 Review of interoperability for CNC manufacturing

4.1 Introduction

This chapter presents a review of interoperability mechanisms for CNC manufacturing. In reviewing the advantages and disadvantages of the existing methods for interoperability between CAx chains, the gaps in the current body of research are identified.

4.2 CAx systems and interoperability

From the design of the first Numerical Controlled machine onwards, engineers have sought to realise the dream of fully automatic manufacture of discrete components (Xu et al., 2005). Numerical Controller (NC) is a control device that machines a target part by activating the servo motor according to commands; when NC is combined with computer technology it is called Computer Numerical Control (CNC) (Suh et al., 2008). The programming language used by these machines was initially similar to the codes that were used for 2-axis scientific plotters made by Gerber Scientific (Schroeder, 1998). With the addition of a similar strategy for including the Z axis to the codes it became possible to support 3-axis milling machines (Schroeder, 1998, Nassehi, 2007). These codes became known as G&M-Codes with “G” denoting movement commands and “M” denoting miscellaneous commands (i.e. Coolant On/Off, Tool change, etc.).

These commands would require low-level planning of the production process and dividing the process plan into a sequence of switching instructions. This work is tedious and in case of all but the most simple components, prone to errors. Efforts were thus made to automate this process.

Computer Aided Design (CAD) systems have allowed engineers to create a digital representation of the geometry of the target part so that Computer Aided

Manufacturing (CAM) systems can then generate the G&M codes for the part automatically (Groover and Zimmers, 1983). CAD, CAM and CNC systems, which together are now known as Computer Aided systems (CAx) (Werner Dankwort et al., 2004), therefore form a continuous chain from the design to the manufacturing of a part.

In most common CAx systems the part design is transferred to the CAD system where its geometric information is defined in CAD files. These CAD files are then transferred to the CAM systems where the tool paths, feeds and speeds are added in. Then, G&M codes are generated using a CAD/CAM post processor and, finally, the result is transferred to the CNC machine controller for machining the part. The current CAx manufacturing chain is illustrated in figure 4.1.

From 1950 to 1955 all designers and engineers drew components manually on drawing paper but in the 1960s with development of computers, they started drawing sketches using computers. Around the same time the first CAD systems were developed. Between the 1970s and the 1980s 3D CAD systems and solid modelling frameworks were introduced. Expert CAD systems and manufacturing were developed in 1990s followed by enterprise manufacturing from 2000. In the late 1990s the concept of Product Life-cycle Management (PLM) also came to the industry from university research and CAD systems redefined themselves to these new concepts.

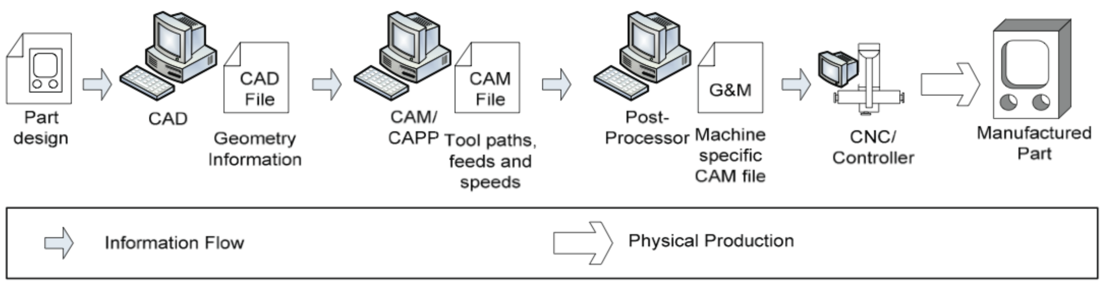


Figure 4.1: The CAx manufacturing chain (Newman and Nassehi, 2007)

A brief history of CAD/CAM system is presented in table 4.1 documenting the major advances in software development.

Process planning acts as a bridge between design and manufacturing by translating design specifications into manufacturing process details (Engelke, 1987). The development of process plans involves the following set of activities:

- Analysis of part requirements;
- Selection of raw work piece;
- Selection of manufacturing operations;
- Selection of machine tool(s);
- Selection of tools, tools holders, work holders;
- Selection of manufacturing conditions (cutting speeds, feeds, depth of cut, etc.);
- Determination of manufacturing time.

In cross technology interoperability the assumption is that in the selection of manufacturing operations and machine tools there are several closely competing solutions. Thus, the process plan may be adapted for various technologies with minimal economic penalties.

Manual process planning is mainly based on a manufacturing engineer's experience (Xu et al., 2011) and knowledge of production facilities, tooling, equipment, etc. The major problem with this approach is that it is time consuming and development of the process plan depends on an individuals expert knowledge and situation.

Table 4.1: History of CAD/CAM

Year	Developments in CAD/CAM software
1950	Part programs prepared manually
1955	MIT started APT development
1960	SKETCHPAD by MIT
1965	UNISURF, Bezier's patch, CAD Drafting
1970	Solid modelling
1975	3D CAD systems
1980	PC based CAD, AutoCAD
1985	MAP, TOP LAN standards
1990	Expert CAD systems, Neural Nets
1995	Virtual Manufacturing
2000	Enterprise Manufacturing
2005	CAD systems with PLM systems

4.2.1 Computer aided process planning (CAPP)

Effort has been made to automate the process planning activity through the use of computer aided process planning (CAPP) systems.

A basic CAPP model by Marri et al (1998) is illustrated in figure 4.2. This shows that the part geometries from CAD systems are prepared for CAPP and with combination of knowledge and physical rules in the CAPP systems and with the use of a post processor, production planning and scheduling information can also be estimated.

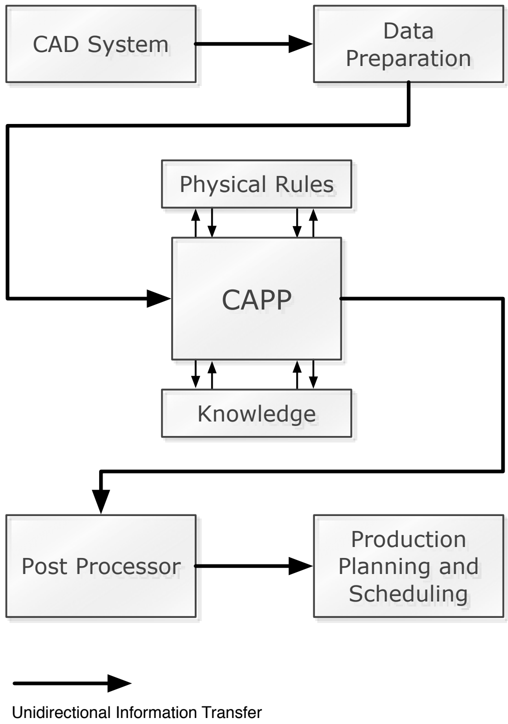


Figure 4.2: Basic CAPP model adapted from (Marri et al., 1998)

In general there are two approaches in CAPP: variant and generative (Marri et al., 1998).

Variant: this approach is suitable for the small manufacturing products, which have a stable manufacturing process. This approach follows the principle that similar parts that require similar plans, it requires the operator to classify the part and process plan for the part from the databases.

The advantage of this approach is that it is easy to maintain the databases. One of the disadvantages of the variant approach is that the quality of a process plan depends on the knowledge and background of the process planner (Xu, 2009, Zhang and Xie, 2007).

Generative: In this approach the process plan is generated automatically with little human involvement for each part. The benefit of this approach is the generation of new process plans based on decision logic and process knowledge for each part. This approach is suitable for big and complex manufacturing products.

One of the advantages of the generative approach is that the level of detail is much greater than the variant approach and more importantly, the quality of the process plan depends less on the knowledge and background of a process planner.

Here, the process planning depends on computer captured knowledge and includes selection of processes, machine tools, fixtures, inspection equipment, tools and classification of operations (Xu, 2009).

As shown in the table 4.2, in the 1960s, using a computer in process planning was first proposed. In the 1980s most of the researchers' work was on understanding of the potential of using CAPP in manufacturing. New technologies for CAPP systems were developed in the 1990s, with many reviews on CAPP research realised the need for Artificial Intelligence (AI) technology for CAPP architecture (Ham and Lu, 1988).

By the end of the 1990s most CAD systems started using CAPP concepts for communicating with CNC machines using feature and feature recognition (Cay and Chassapis, 1997).

In the 2000s the developments in CAPP systems were based on new technologies such as Virtual reality-based process planning (Peng et al., 2000), Neural network-based process planning (Yue et al., 2002) and Expert systems-based process planning (Gupta and Ghosh, 1989, Kiritsis, 1995, Liao, 2005, Metaxiotis et al., 2002).

Table 4.2: Review papers for CAPP

Year	Authors	Description of the work
1965	(Niebel, 1965)	The initial idea of process planning using computers.
1984	(Steudel, 1984)	Comparing CAPP and traditional process planning methods. The results indicated the superiority of the traditional methods at the time of writing.
1985	(Eversheim and Schulz, 1985)	Focused on the historical evolution of assembly process planning.
1988	(Ham and Lu, 1988)	The first comprehensive review of CAPP featuring 200 publications and 14 distinct CAPP systems.
1989	(Alting and Zhang, 1989)	A review of 128 CAPP system proposing categorisation into variant, semi-generative, generative and expert process planning.
1991	(Shah et al., 1991)	CAD and feature based system and their relation ship with CAPP is explored.
1993	(ElMaraghy et al., 1993)	Considered the effect of CAPP on quality and standards.
1993	(Eversheim and Schneewind, 1993)	Assembly planning, function integration with NC programming, using of AI methods in decision making and using of database sharing for data integration with CAD.
1995	(Kamrani et al., 1995)	An overview of techniques and the role of process planning.
1996	(Leung, 1996)	A review of 200 publications highlighting the potential role of Artificial Intelligence in computer aided process planning.
1997	(Cay and Chassapis, 1997)	Presenting the notion of manufacturing features and feature recognition technologies.
1998	(Marri et al., 1998)	The latest developments in CAPP together with advantages and disadvantages of the competivs systems were discussed.
2002	(Yue et al., 2002)	Investigated the use of neural networks in computer aided process planning as reported in the literature.
1989/1995 2002/2005	(Gupta and Ghosh, 1989, Kiritsis, 1995, Metaxiotis et al., 2002, Liao, 2005)	A series of reviews on the use of expert systems in manufacturing planning with a focus on process planning.
2006	(Shen et al., 2006)	Presenting state-of-the-art on agent-based distributed manufacturing process planning and scheduling.
2007	(Zhang and Xie, 2007)	Providing a review of agent technology for collaborative process planning.
2011	(Xu et al., 2011)	A critical review of recent developments and future trends on Computer-aided process planning.

The early CAPP systems were often an intellectual exercise in assessing the capabilities of then emerging computer systems in facilitating manufacturing decision making. Limited in scope and capacity, these system were shunned by the industry. The pivotal development which rekindled the interest in CAPP was the introduction of feature based technologies.

With a limited set of conceptual constructs, feature based systems lowered the technical requirements for artificial intelligence in CAPP to a level that is practically achievable with modern computing technology. Within the scope of this research, the focus is thus is on developments after these events.

Since then many technologies have been proposed for CAPP systems: Knowledge-based systems, Neural networks, Genetic algorithms, Fuzzy set theory/logic, Petri nets, Agent-based technologies, Internet-based technologies, STEP-compliant CAPP leading to latest developments such as energy efficient CAPP (Lu et al., 2006, Xu et al., 2011). A review of these related technologies is given below:

i Featured-based technologies in CAPP

Almost all modern CAPP systems function on the basis of features. There are two approaches related to feature-based technology: feature recognition and design by features. Feature recognition examines the geometry and topology of the parts to determine the definition of features.

There are two distinct methodologies for design by features and they are: destruction by machining features and synthesis by design features. The former methodology starts from raw stock and describes which parts should be machined; a design model is generated from subtracting materials by machining volumes. The latter builds up the part by describing volumes of material that make up the part (McMahon and Browne, 1993).

ii Knowledge-based technologies in CAPP

The knowledge-based system gives a process planning system an adaptive and learning capability, the basic advantages of these systems are self learning and intelligence (Tsatsoulis and Kashyap, 1988). Expert systems are among the most common implementations of knowledge-based technologies.

Expert systems usually consist of three main components; the knowledge-base, the inference engine and the user interface. Experts systems organize knowledge in rules and control strategy, which allow users to modify a program with ease.

They are able to organize knowledge in such a way that they can reason intelligently and are designed to deal with complicated problems such as process planning (Grabowik and Knosala, 2003).

The major problems in using expert systems are the quality of the knowledge, the performance of the inference engines namely the non-determinism involved in reasoning about complex problems and difficulties in tracing potential errors in the decision making process (Shu-Hsien, 2005).

Table 4.3 lists a number of papers where the authors applied expert systems to solve process planning problems. The simpler variation of these systems are those that use "if then else" rules to form a rudimentary knowledge-base without the capability of automatic generation of new knowledge.

The scope of this research and the focus on specific operations, make such systems appealing candidates for development of potential solutions as there is little ambiguity in the context and a limited number of concepts within the system.

Table 4.3: Knowledge-based CAPP systems

Year	Researcher/s	Development	Description
1997	(Younis and Abdel Wahab, 1997)	Development of an expert CAPP system for rotational components	Based on analysis of process planning domain in terms of problem specifications and knowledge characteristics.
1998	(Kryssanov et al., 1998)	Proportion of a formal method for designing a CAPP expert system	Description of theoretical aspects of the formalism.
1999	(Jiang et al., 1999)	Creation of an Automatic Process Planning System (APPS)	Generation of manufacturing process plans directly from CAD drawing.
2000	(Arezoo et al., 2000)	Expert Computer Aided Cutting Tool Selection (EXCATS)	To selection of cutting tools (including tool holders and inserts) and conditions of turning operations (feed, speed and depth of cut)
2001	(Pham and Gologlu, 2001)	Designing of hybrid CAPP system (ProPlanner)	For facilitating concurrent product development.
2002	(Anwer and Chep, 2002)	Creation of an Intelligent Process Planning Assistant (IPPA)	Supports knowledge-assisted planning for machining operations and presented an opportunity for CAD/CAM integration.
2002	(Zhao et al., 2002)	Further extended work on EXCATS system	Integrating EXCATS with a CAD system.
2004	(Gologlu, 2004)	Extended the ProPlanner,	Enables the problem of operation sequencing to be systematically addressed.

iii Genetic algorithm technologies in CAPP

A genetic algorithm is an intelligent search method to solve a problem without much data about overall mechanisms in the system (Holland, 1992). The advantage of using this approach in CAPP is that the selection of machine tools, cutting tools, tool access direction for each operation and movement among the operation take place concurrently (Rocha et al., 1999). This makes it possible to find a globally optimal process plan for a part. Genetic algorithms are often combined with other methods to solve optimization problems in generative CAPP. A number of researchers have used genetic algorithms for estimating cost of production based on a selection of machining parameters such as the number of passes, depth of cut in each pass, speeds and feeds (Shunmugam et al., 2002, Shunmugam et al., 2000). With the focus of this research on interoperability, the link between features and manufacturing operations already exist and need to be transformed rather than generated. The generative capabilities of algorithms such as genetic algorithms are thus not necessary.

iv Internet-based technologies in CAPP

Enterprise strategy in manufacturing environments and cooperative process planning among engineers at different places, have led to the developments of internet-based CAPP technologies. Table 4.4 summarises a number of papers in the field.

v STEP-compliant technologies in CAPP

With the need of manufacturing for sharing information between different applications including CAPP, the International Organisation of Standardisation (ISO) decided to develop a data model standard to support the exchanging and sharing product information. The model called Standard for Exchange of Product Model Data (ISO 10303-11, 1994, Xu et al., 2011) better known as STEP, provides a universal format for CAD geometry through a neutral interface.

4.2.2 CNC programming

Throughout the development of multi-axis CNC machines, the NC programming language has remained basically the same with G&M code programming, and this has caused a lack of interoperability between systems from the CAM stage to the manufacturing stage (ISO 6983-1, 1982, ISO 10303-11, 1994).

Table 4.4: Researches in internet-based technology

Year	Researcher	Development	Technology
1998	(Feng and Zhang, 1998)	An internet enabled CAPP system based on the COM distributed model.	COM
2001	(Qiu et al., 2001)	Distribution of multi-user environment on the internet.	EAI, Java
2001	(Sun et al., 2001)	Presenting an agent-based concurrent engineering system concerning product design and manufacturing planning.	Agents
2004	(Chung and Peng, 2004)	Development of Web-based tools and machine selection system (WTMSS)	VRML, EAI, Java
2005	(Peng et al., 2005)	Developments of an Internet-enabled system for setup planning in machining operations.	Java, Web, XML
2008	(Alvares et al., 2008)	Integrated Web-based CAD/CAPP/CAM system for the remote design and manufacture of feature-based cylindrical parts.	Java, HTML
2008	(Hu et al., 2008)	Development of XML-based implementation of manufacturing routs sheet documents for context-sensitive and Web-based process planning.	XML
2009	(Agrawal et al., 2009)	Development of a multi-agent system for distributed process planning	Agents, XML

Currently, in a CAx chain, all the information is transferred in a unidirectional manner from design to manufacturing. In case of any changes in the code on the CNC controller by the operator on the shop floor, the integration of the information from controller to postprocessor is not maintained because of the unidirectional information flow from postprocessor to controller.

Within an interoperable manufacturing network, however, each system has the capability of transferring information to any other system, including from a CAM system to a CAD system. Figure 4.3 illustrates the differences between a CAx chain and an interoperable CAx network.

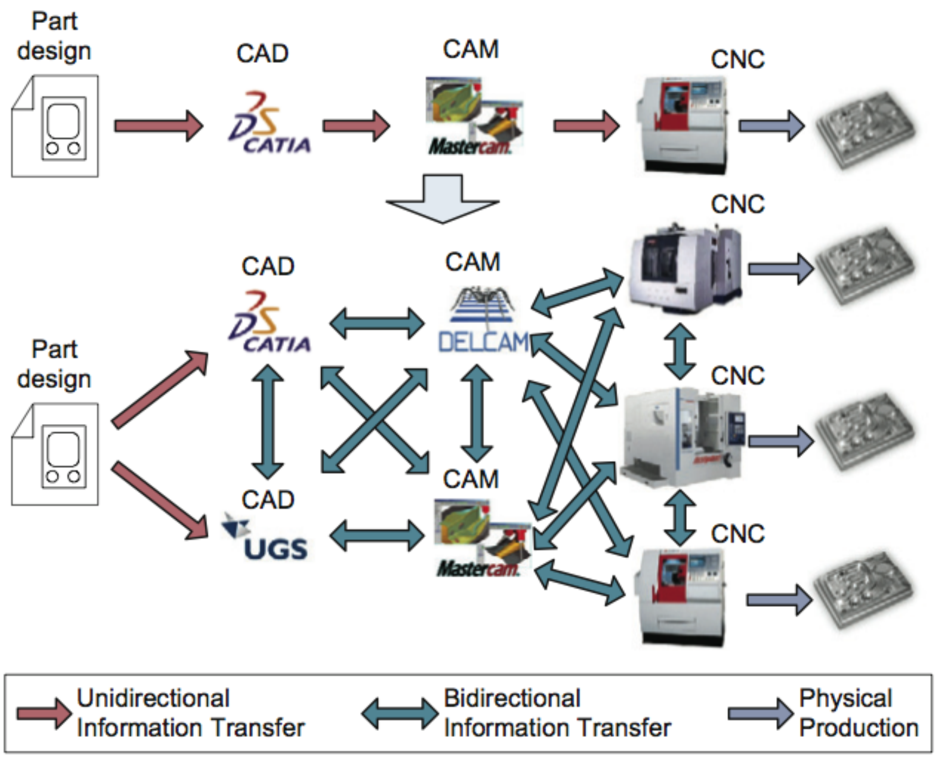


Figure 4.3: An interoperable CAx network (Newman and Nassehi, 2007)

As indicated in section 2.5 the focus of this research is on enabling interoperability between CNC machines with different technologies. After process planning, a part program for controlling the CNC machine tools is generated. This generation can be done by the manual programming method or the automatic method (Suh et al., 2008).

In manual part programming, programmers directly edit the program in CNC readable G&M-codes. In the automatic programming method, the programmer edits the program in terms of graphical symbols or a high-level language via a computer and then the CNC system converts these programs into machine-readable instructions. The various programming methods as categorised by Suh et al. (2008) are shown in figure 4.4.

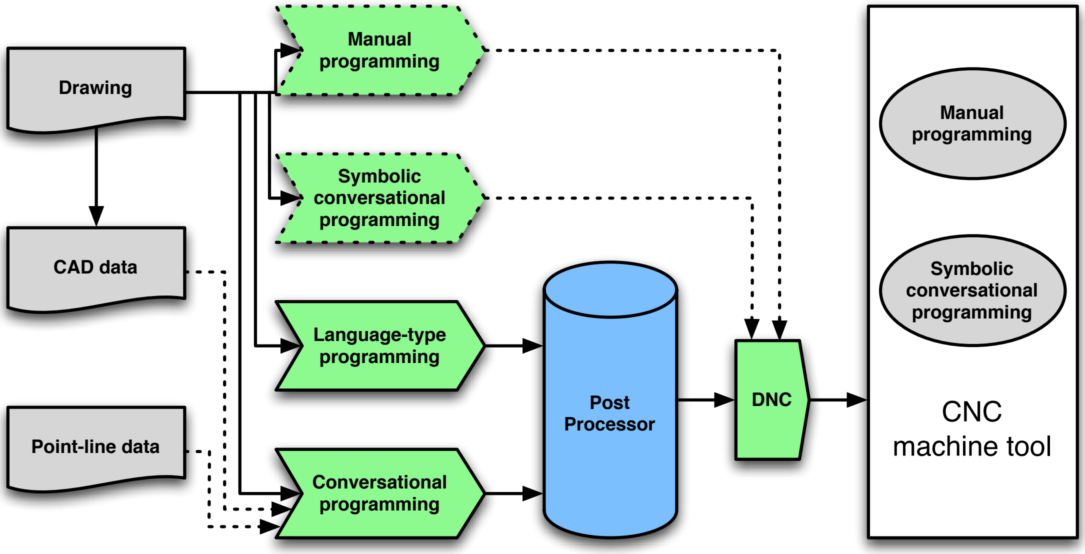


Figure 4.4: Programming methods adapted from (Suh et al., 2008)

4.3 CAx interoperability standards

Through the development of the CAx process chain, standards were developed for machining to solve the bottleneck of data format and data integration between machines and systems. A review of standards related to interoperability is summarised below.

4.3.1 IGES, VDA-FS

The introduction of the Initial Graphics Exchange Specification (IGES) was the beginning of the development of standards for exchanging geometry information; this standard was developed in 1979 by a group of CAD users and vendors (Parks, 1984). One of the disadvantages of this standard was its limitation in free-form surface data representation. This shortcoming was subsequently addressed by the development Verband der Automobilindustrie - Flächenschnittstelle (VDA-FS), which was proposed in 1999 by VDA to focus on free-form surfaces, specifically for representing data for geometries in use by automotive manufacturers. The specific focus of this standard inhibited its wide adaption, leaving a gap for a universal CAD standard (Strasser and Seidel, 1989).

4.3.2 The Standard for the Exchange of Product DATA Model (STEP)

The successor to IGES and VDA-FS was developed to fill this gap, this successor, standardised as the standard for exchange of product data (STEP) ISO10303 provided a computer-interpretable definition of the physical and functional characteristics of a product throughout its lifecycle. The second phase of the development started in 2002 to expand the standard's capability in a wider range of industries such as aerospace, automotive, electrical and electronic. The third phase, with the introduction of the STEP modular architecture, solved the problems resulting from massive STEP schemas and dealing with very large file sizes (Pratt, 2004).

4.3.3 STEP-NC

With the wide adoption of STEP, researchers proposed a similar standard to be developed for storing process plans at the level of operations, features and strategies. This emerging standard developed for physical control of CNC controllers was written using the data modelling language EXPRESS (ISO 10303-11) and standardised as ISO 14649.

It was then integrated with the geometry data model of STEP to form an application protocol (a protocol for assigning extra semantics to data entities) within the STEP suite of standards as ISO 10303-238. The two standards, ISO 14649 and ISO 10303-238 are collectively known as STEP-NC.

The vision for the development of these standards was to revolutionise the CAD/CAM/CNC chain to the paradigm shown in figure 4.5. The aim was to remove toolpath generation and post processing from the CAM system (some researchers consider the resulting system a CAPP system) and pushing these activities to the

CNC controller. The resulting “intelligent” controller would thus be able to read a list of operations and related features and interpret these according to machine capabilities, generate the toolpaths and control the servos to produce the part with given tolerances.

The development of STEP-NC started in 1999 with ESPRIT and IMS working together with Siemens and the Universities of Aachen and Stuttgart in Germany. Komatsu and Fanuc in Japan, Heidenhain in Switzerland and Pohang University of science and technology in Korea extended this work (Suh et al., 2002).

The first models for controlling CNC milling and turning were presented in 2005 with draft models for wire and die-sink electro discharge machining (EDM) being developed in parallel (ISO 14649-11, 2002, ISO 14649-12, 2004). In 2006, STEP-NC AP238 was published by ISO (ISO 10303-238, 2006).

From the end of 2006 to the beginning of 2007 a live 5-axis STEP-NC machining demonstration was hosted by Airbus, with further modelling and measurement demonstrations being conducted in Japan (Krzic et al., 2009).

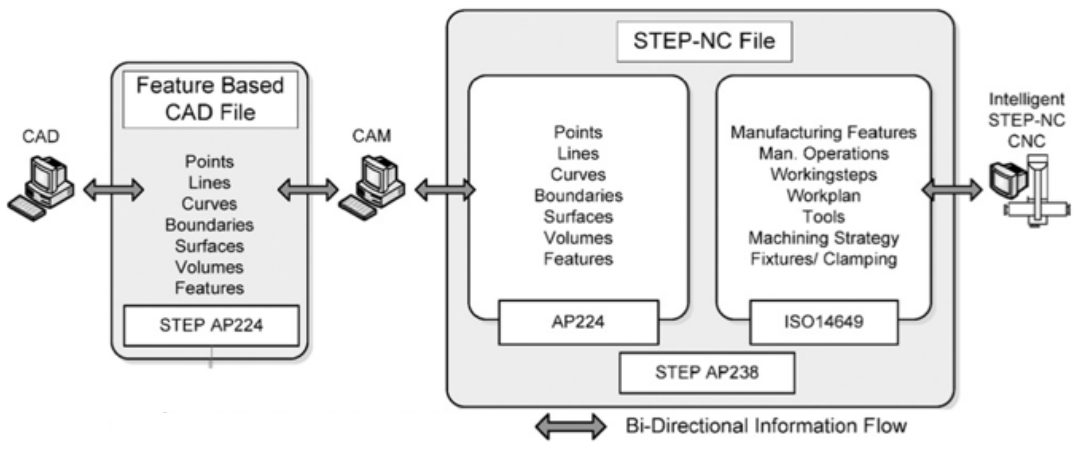


Figure 4.5: Information flow in a STEP-NC CAD/CAM/CNC chain
(Newman et al., 2008)

A process for using STEP-NC for feed and speed optimization, high-speed machining, tolerance driven tool compensation and traceability was developed in Sweden in 2008. At the same time, ways of using STEP-NC for closed loop machining, feed optimization and measurement were being discussed (ISO-TC184-CS4-WG3-T24, 2009).

In 2009, the machining of example mould test parts were undertaken on different machines, and in 2010 machining of a part in multiple steps with multiple CNC machines, such as 3,4, and 5-axis milling machines was tested to enable STEP-NC to be used for complex parts with multiple set-ups.

In 2012, a crown wheel gear was machined on a milling CNC centre to demonstrate the STEP-NC machining accuracy by updating the information in STEP-NC machining data with predicted data from cutting tool under load. The lists of ISO14649 (STEP-NC) parts are shown in the table 4.5.

Table 4.5: List of STEP-NC parts

Part	Description
ISO 14649 Part 1	Overview and Fundamentals
ISO 14649 Part 10	Generic process data
ISO 14649 Part 11	Process specific data for Milling
ISO 14649 Part 12	Process specific data for Turning
ISO 14649 Part 13	Process specific data for Wire-EDM
ISO 14649 Part 14	Process specific data for Sink-EDM
ISO 14649 Part 16	Inspection
ISO 14649 Part 111	Tools for milling
ISO 14649 Part 121	Tools for turning
ISO 14649 Part 201	Machine tool data for cutting process

ISO 14649 is comprised of various parts each dealing with specific technology, related parts to this research are:

i. ISO 14649 part 1: Overview and fundamental principles

The basis of the ISO 14649 was introduced in this part by describing the context of the standard, and overview of the architecture and the development plan. Generally, STEP-NC code is comprised of two main sections. The first section is a header that gives some general information such as filename, author, date, organization, etc. The second section is the main section of the program that contains information about the geometry, features and manufacturing tasks.

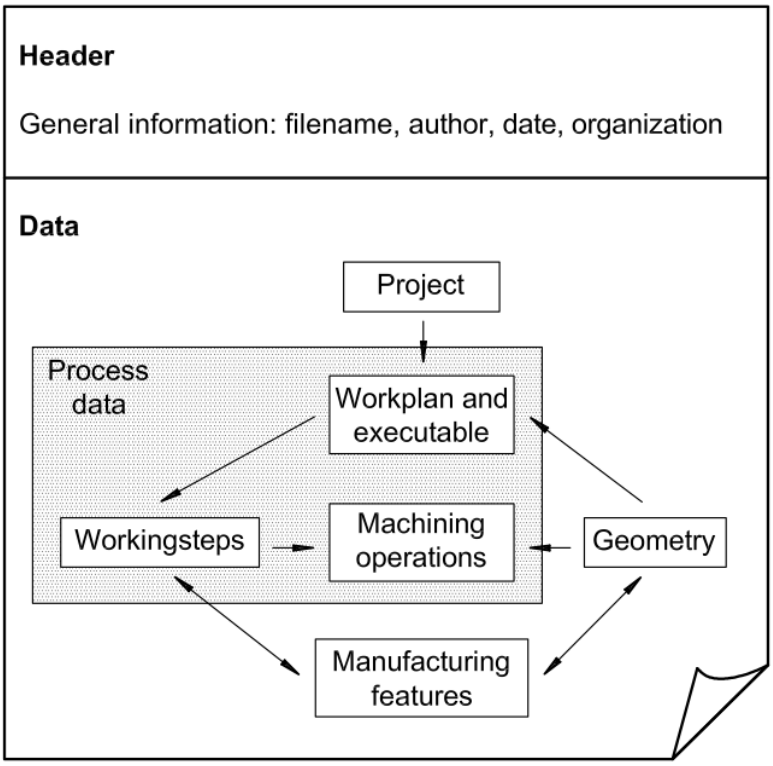


Figure 4.6: Data structure of ISO14649 (ISO 14649-1, 2002)

Figure 4.6 shows the basic structure of a STEP-NC (ISO 14649) file. Each STEP-NC file has a single `PROJECT` entity which serves as a starting point for the data. The project then specifies a `workplan` consisting of executable `working_steps` which describe the `machining_operations` and `machining_features` on the part. Figure 4.7 shows the top level hierarchy of features and operations and their relationship within the content of STEP-NC.

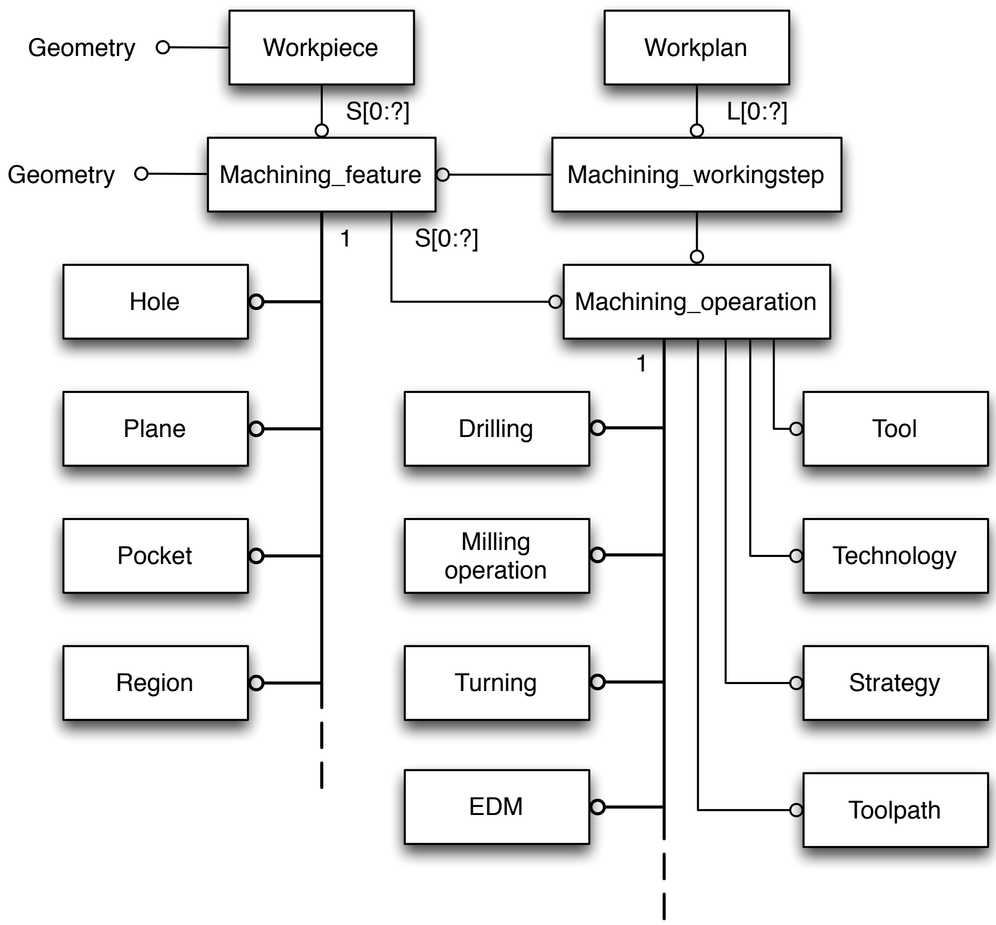


Figure 4.7: Overview of ISO 14649 data model structure adapted from (ISO 14649-1, 2002)

ii. ISO 14649 part 10: General process data

This part specifies the process data which is needed for programming NC code for all machining technologies. The data model for manufacturing features and the control structure of the program is defined in this part of the standard.

iii. ISO 14649 part 11, part 12: Process data

These two parts specify the data model for machine technology for milling operations (ISO 14649-11, 2002) and turning operations (ISO 14649-12, 2004), they generally focus on the entities representing the manufacturing process.

iv. ISO 14649 part 111. Part 121: Cutting tools for milling and turning operations

Cutting tools for milling and turning operations are described in these parts, with ISO 14649 part 111 working with ISO 14649 part 11 for milling operations (ISO 14649-111, 2002) and ISO 14649 part 121 working with ISO 14649 part 12 for turning operations (ISO 14649-121, 2004).

4.4 The state of the art in manufacturing interoperability

Interoperability research started with the development of intelligent controllers as they were seen as a major enabler. One of the earliest works in this area was undertaken by Shimamura in 1996 was the development of NC machines that could economically use a PC-based retrofitting scheme for the manufacturing of free-form surfaces (Shimamura et al., 1996, Newman et al., 2008). In 2002, Suh and Cheon began to implement developments in manufacturing technology and processes in terms of proposing a framework for intelligent CNC systems (Suh and Cheon, 2002).

This framework was followed by Hardwick as a first attempt at STEP-NC (Hardwick, 2002). Research on a five-axis milling machine that uses STEP-NC was designed by Lee and Bang (Lee et al., 2006) and another prototype was proposed by Newman (Newman et al., 2003) as a STEP-Compliant CAD/CAM system using ISO

14649. A number of researchers have worked on STEP-Compliant process planning in manufacturing, as identified by Xu et al. (Xu et al., 2006). Kumar and Nassehi (Kumar et al., 2007), meanwhile, have introduced an intelligent and self-learning framework that updates the STEP-NC information based on feedback from on-machine measurement.

In 2007, Amaitik and Engin Kilic (Amaitik and Engin Kilic, 2007) achieved a successful system, which they called ST-FeatCAPP, which was based on prismatic parts. Further research was carried out by Liu on prismatic parts covering a complex feature recognition process, which enabled generation of the corresponding machining operation via the use of STEP-NC standards (Liu et al., 2006).

Since 2004, researchers have started proposing frameworks for interoperability of CNC machines, primarily in respect to turning machines (Xu and Wang, 2004, Yusof et al., 2006). Shin in 2007 developed a system which had the capability of translating G&M codes to STEP-NC for turning operations (Shin et al., 2007). In 2009, Yusof developed a computational environment for a STEP-NC compliant system for turning (Yusof et al., 2009). Zhang in 2010 developed a new software tool to demonstrate the feasibility of interoperable CNC manufacturing based on STEP-NC (Zhang et al., 2010). In 2012, a system was developed to read G&M codes and generate STEP-NC file for milling technology by Zhang (Zhang, 2012). An overview of this interoperable research work relating to turning and milling machines is shown in table 4.6.

Interoperability in manufacturing using CNC machines has been a major area of research during the last ten years, with some of the significant works in this area listed in table 4.7. A group of researchers in 2005 reviewed the development of STEP-NC for a range of CNC processes and tries to show a view of STEP-NC applications for CAD, CAPP, CAM and CNC integration (Xu et al., 2005).

Table 4.6: STEP-Complaint systems

Year	Systems	Technology			Standard	
		Mill	Turn	Other	Input	Output
2003	SFPS (Suh et al., 2003)	☑			ISO 10303	ISO 14649
2004-2005	G-Code free (Xu et al., 2005, Xu and Wang, 2004)		☑		ISO 10303	Native CNC language
2006	STEPTurn (Heusinger et al., 2006, Xu, 2006)		☑		ISO 10303	ISO 14649
2006	TurnSTEP (Choi et al., 2006, Suh et al., 2006)		☑		ISO 10303	ISO 14649
2007	G2STEP (Shin et al., 2007)		☑		G-Codes	STEP-NC
2009	SCSTO (Yusof et al., 2009)		☑		STEP Part12	STEP-NC
2010	PPS (Zhang et al., 2010)		☑		STEP-NC	STEP-NC
2012	UPCi (Zhang, 2012)	☑			ISO 10303	STEP-NC

Xu has provided a comprehensive review of the technology relating to the futuristic use of STEP-NC to support distributed interoperable intelligent manufacturing through global networking with autonomous manufacturing workstations with STEP compliant data interpretation, intelligent part program generation, diagnostics and maintenance and monitoring of job production scheduling (Xu and Newman, 2006). In 2007 Nassehi outlined the barriers and issues in regard to incompatibility between the various CAD/CAM/CNC systems and proposed a new framework to overcome these barriers in achieving interoperability in the CAD/CAM/CNC chain (Nassehi et al., 2007b).

Table 4.7: Research with a focus on interoperability

Date	**Authors**	**Topic**
2005	Xu, X., H. Wang	The search for intelligent CAD/CAPP/CAM/CNC integration
2006	X.W. Xu and S.T. Newman	A review of the technology in interoperability
2007	A. Nassehi and S.T. Newman	Toward interoperable CNC manufacturing
2008	S.T. Newman, et al	Strategy advantages of interoperability for global manufacturing using CNC technology
2009	Y. Yusof, et al	Interoperable CNC system for turning operations
2010	Zhang	A STEP-compliant process planning system for CNC turning operations
2012	Y.Yusof et al	Review of STEP-NC compliant system for turn-mill operations
2013	X. Zhang, et al	Process comprehension for shopfloor manufacturing knowledge reuse

Newman in 2008 provided a strategic view of how interoperability can be implemented across the CAx chain with the range of standards used to regulate the flow of information (Newman et al., 2008). In 2009, Yusof described and illustrated a STEP-compliant CAD/CAPP/CAM system for the manufacturing of rotational parts on CNC turning centres (Yusof et al., 2009).

In 2010, Zhang developed new software tools to demonstrate the feasibility of interoperable CNC manufacturing based on STEP-NC (Zhang et al., 2010). Yusof reviewed the computational environment for a STEP-NC compliant system for turn-mill operations in 2012 (Yusof and Nor, 2012).

Zhang developed more recently, a new vision of the shopfloor interoperability associated with process knowledge capture and reuse (Zhang et al., 2013).

There are other methods for interoperability in manufacturing such as agent technology which has been utilised in interoperable CAx manufacturing research by a number of researchers.

Wang et al (1998) introduced the next-generation of intelligent manufacturing systems to control machines in an interoperable manner by using function blocks to design the next generation of manufacturing control systems. Zhou and Jiang, used mobile agents to encapsulate manufacturing resources to allow them to communicate over the internet (Zhou et al., 2005).

Computer aided process planning is a major topic of research in the CAx manufacturing domain. In 1996 by using commercial CAD packages and a commercial process planning system, researchers explored direct interfacing of CAD systems with process planning software (Srinivasan and Fischer, 1996).

Wang et al utilised function blocks as a reusable coding device to encapsulate process plans by combining the reusable basic function blocks properly which gives a machine more intelligence and autonomy to make decisions on how to adapt its behaviour to actual ongoing machining processes (Wang et al., 2006). Another method for interoperability is based on enterprise integration. There are three approaches in manufacturing enterprises: integration by merging functions of different sections, integration through a distributed or centralised databases for production information and integration through the establishment of linkage mechanisms between functions (Twigg et al., 1992).

Another method for interoperability is semantic interoperability; this method ensures that the same set of semantics is interpreted from data that is transferred between two systems (Nassehi, 2007).

A number of researches have identified open and intelligent CNC controllers as the main enabler for interoperability. Pritschow recognised open control systems as a key enabler for modular and re-configurable manufacturing systems (Pritschow et al., 2001, Nassehi, 2007).

Web-enabling of manufacturing resources as a means to better integrate them in a global enterprise leading to better interoperability is another method for interoperability in the manufacturing chain. The ability to monitor and control resources on the internet is the one of the main foci in web-enabling of manufacturing (Wang et al., 2003).

Cloud manufacturing is a new evolving area toward interoperable manufacturing by enabling online tracking or resource utilization, multi-objective decision support for process planning and simulation, and on-board manufacturing execution control (Wang and Bi, 2013).

As mentioned in the scope of research (see section 2.5), the focus is achieving interoperability between CNC machine tools with different technologies without using any CAD/CAM systems. The aforementioned methods all rely on CAx and hence although they have merit, they are not appropriate for this research.

4.5 Methods for achieving interoperability between CNC machines

There are two fundamental approaches for manufacturing parts with new machines. These are:

i. Redesigning the part with the CAD system from the beginning for the new machine.

ii. Converting NC code from an old machine to a part programme for execution on the new machine.

In the first method (redesigning the part from the beginning) the designer should make a new drawing from the old part geometry, import the information to a CAD system, and then regenerate NC codes for the new CNC machine using a new CAM system and postprocessor that works with the new machine.

This method is costly and time consuming and, if the part needed to be machined with different machines in the future, the company would need to regenerate the CAD geometry and make manufacturing decisions through CAM to the postprocessor again.

Another disadvantage is the need for original design information in an accessible format. This method is illustrated in figure 4.8.

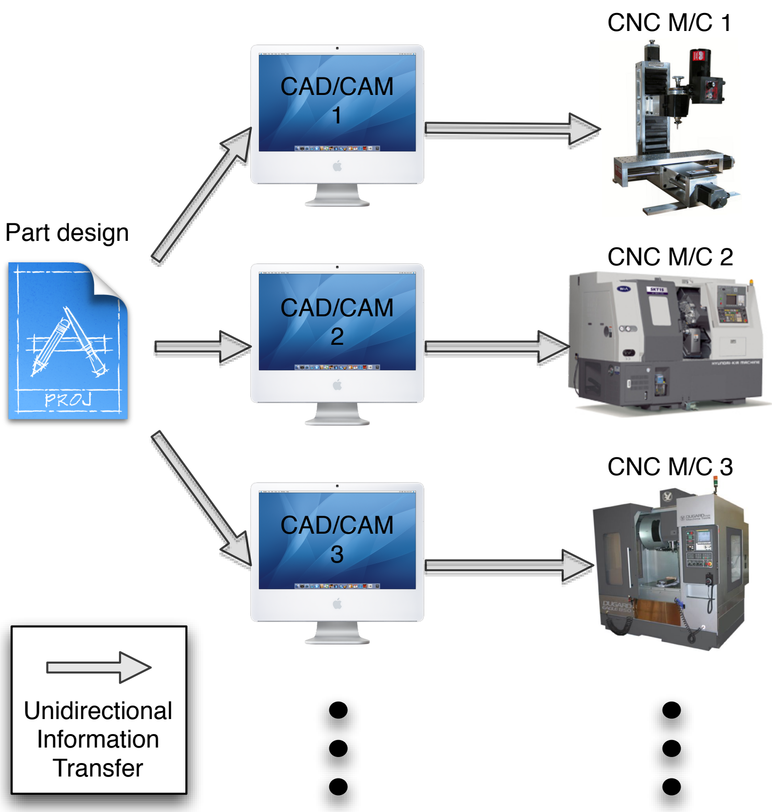


Figure 4.8: Redesigning method

Figures 4.9 illustrate the manufacturing information flow in the state-of-the-art CAD/CAM/CNC. This chain in which the flow at information is predominantly unidirectional, starts with the product designer capturing geometrical information about the product in a CAD system.

The information may be stored as a set of points, lines and curves, or in the case of modern CAD systems, surfaces, boundaries and volumes. CAD systems then store this information in files that are either structured using their proprietary formats or one of the existing standards, such as STEP, IGES or DXF.

The CAD file is then loaded into a CAM system where the manufacturing engineer goes through process planning of the production of the part and decides which tools, operations and machining strategies are appropriate for making the part. The CAM system then generates the toolpaths required to execute the process plan.

The geometry from CAD together with this additional information is stored in a CAM file. At the moment, the file format for storing CAM information has not been standardised and therefore each CAM vendor uses their own proprietary file format.

The toolpaths are then separated from the rest of the data and sent to a post-processor which formats the data in the syntax required to control a specific machine tool. The generated NC file would include only axis movement instructions, switching commands (e.g. coolant on/off), tool changes and such low level commands.

There remains, however, a lack of interoperability standards to enable CAD/CAM/CNC chain to transfer information bi-directionally with each other. Thus there is a need to enable CAD/CAM/CNC systems to convert data from one manufacturing process to another. Currently, this process is highly time consuming, does not provide robust outcomes, and is therefore both impractical and uneconomic.

In the second method, G&M codes are interpreted together with tools and workpiece information to produce a high-level process plan stored as a STEP-NC file. The process is carried out by a software adapter that reverse engineers the G&M codes to convert toolpaths to feature-based operations. This method was first proposed by Shin in developing the software that generates the geometry and process plan data in STEP-NC from G&M codes for turning applications (Shin et al., 2007). Functional architecture of their proposed system, entitled G2STEP is shown in figure 4.10.

The G2STEP system accepts the G-codes, the tool information and the schema detailing the semantics of the language used to write the codes as mandatory input. If available the CAD drawing of the part and the geometry and material data about the raw workpiece are also fed into the system.

The tool information is then used to populate the relevant ISO 14649-112 compliant data model, this is used in conjunction with the other input data in the pre-processor to determine machining functions and technologies relying on the generated tooling information, machining functions and machining technology information.

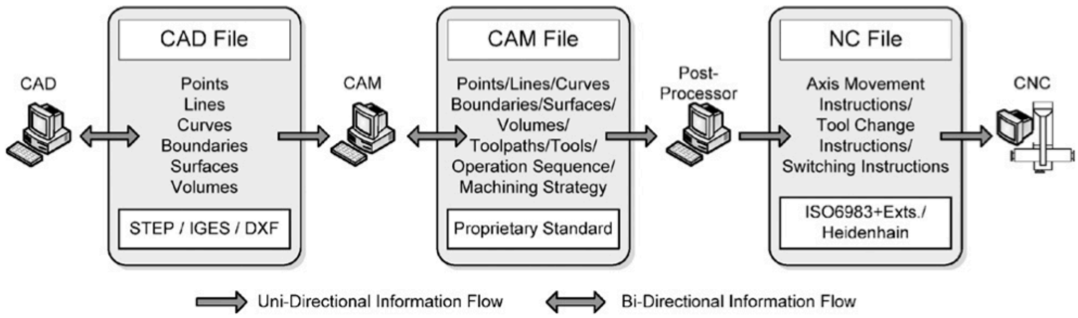


Figure 4.9: Manufacturing information flow in the state-of-the-art CAD/CAM/CNC chain (Newman et al., 2008)

The operations are identified and used as a basis to determine the features and strategies.

Finally, these items of information are put together in the STEP-NC (ISO 14649-10, 12) format and the CAD file (if it exists) is used to confirm the correct execution of the process.

The advantages of this method are:

i. The early adoption of a STEP-Manufacturing environment can be facilitated because a STEP-NC code can be generated easily by operators skilled in using G&M code.

ii. The generated STEP-NC code can be adapted for use on other machines thus facilitating interoperability.

iii. The advantages of rich data models in STEP-NC for storing manufacturing data.

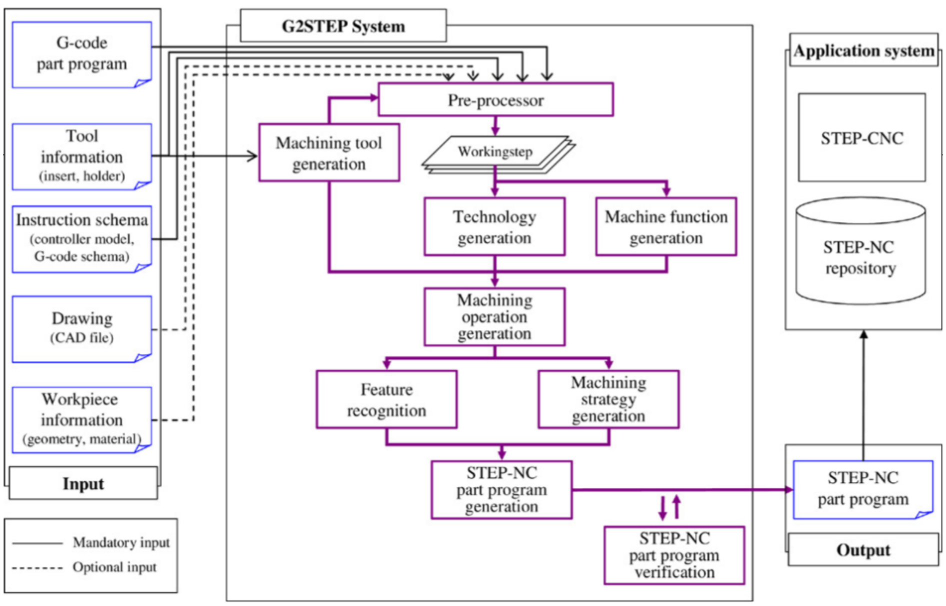


Figure 4.10: G2STEP system functional architecture (Shin et al., 2007)

The Universal Process Comprehension interface (UPCi) is a reverse post process for generating STEP-NC code from G&M code which was developed by Zhang (Zhang et al., 2012). In this approach, the system checks the G&M code syntax and, recognises features by reading the codes and having workpiece geometry and cutting tools information; it then uses ISO 14649 standards to generate the STEP-NC file. Figure 4.11 illustrates the information flow in the UPCi system.

In 2010 Tolio et al. provided a method to identify the machine operations automatically for CNC machine by identifying features through analysing G codes to calculate the removal volume of material by means of moving the cutting tools against the raw workpiece (Tolio et al., 2010).

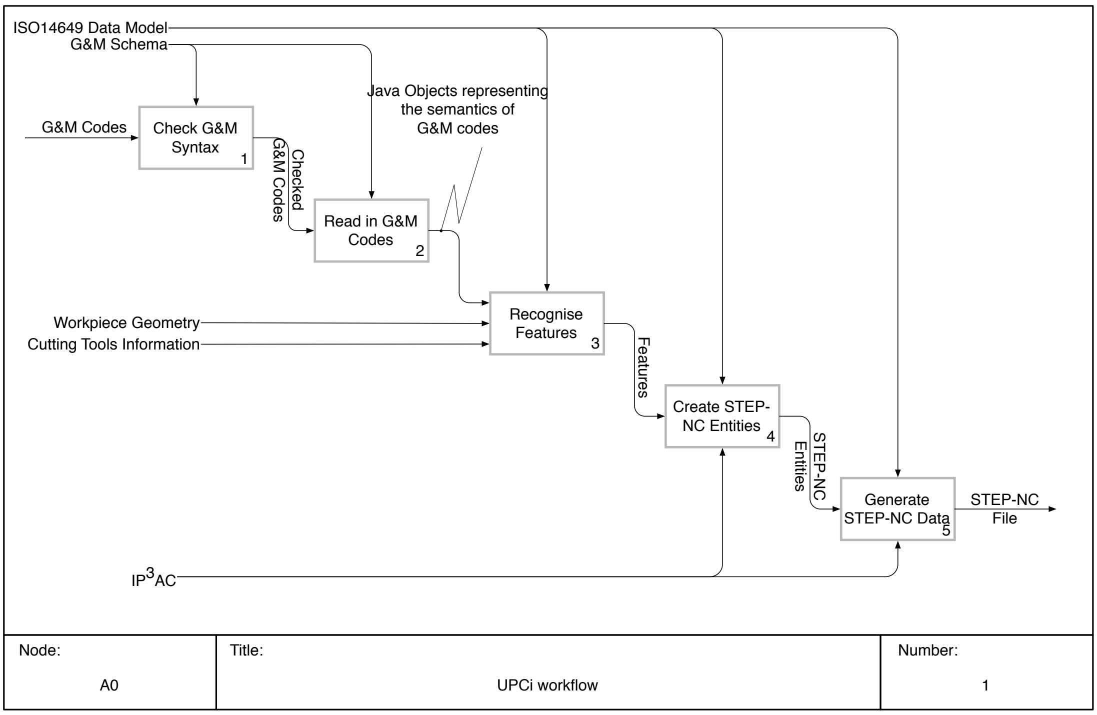


Figure 4.11: IDEF0 view of UPCi workflow adapted from (Zhang et al., 2012)

5 A novel framework for the implementation of cross-technology interoperability for CNC manufacturing

5.1 Introduction

In order to meet the objectives of the research as identified in chapter 2, the systems engineering approach is utilised within this chapter to specify an interoperability framework. First, an overview of the approach is provided followed by an analysis of the requirements for the interoperability system. The functional analysis of the system then follows. The design synthesis of a prototype implementation and verification of the prototype are then presented in chapters 6 and 7.

5.2 An overview of the systems engineering approach

Systems engineering provides a methodology for converting a set of requirements for particular actions into the description of a system to meet those requirements (Anon, 2001). The fundamental activities that form this approach (as seen in Figure 5.1) are:

- **Requirement Analysis:** In this step, the needs that should be fulfilled by the system are analysed and formalised to specify the exact scope for the operation of the system. The output from this phase is a clear definition of the requirements that forms the basis for functional analysis. In this research, the output of the requirements analysis is formatted as the set of inputs to the systems, the desired outputs and the limitations within which the system would be expected to function.

- **Functional Analysis:** After compiling the set of formal requirements, the top level function of the system (which is converting the inputs into the desired outputs) is decomposed to a hierarchy of simpler but more numerous functions. The decomposition process continues until the functions are sufficiently simple to enable design synthesis. The requirement loop ensures that when lower level functions are carried out the overall goal of the system will be reached.

- **Design Synthesis:** In this step, as it would apply to the interoperability framework, the software components that perform the low-level functions identified through functional analysis are specified. The design loop ensures that the specified software components are capable of performing the desired functions identified in the functional analysis. The synthesis of a prototype software system for enabling the cross technology interoperability functionality is documented in chapter 6. The verification loop ensures that the designed software system meets the overall goals of the system. The verification of the prototype system is presented in chapter 7.

- **System Analysis Control and Balance:** This activity is carried out in parallel throughout the entire systems engineering process, includes evaluations of various alternatives and measuring the progress.

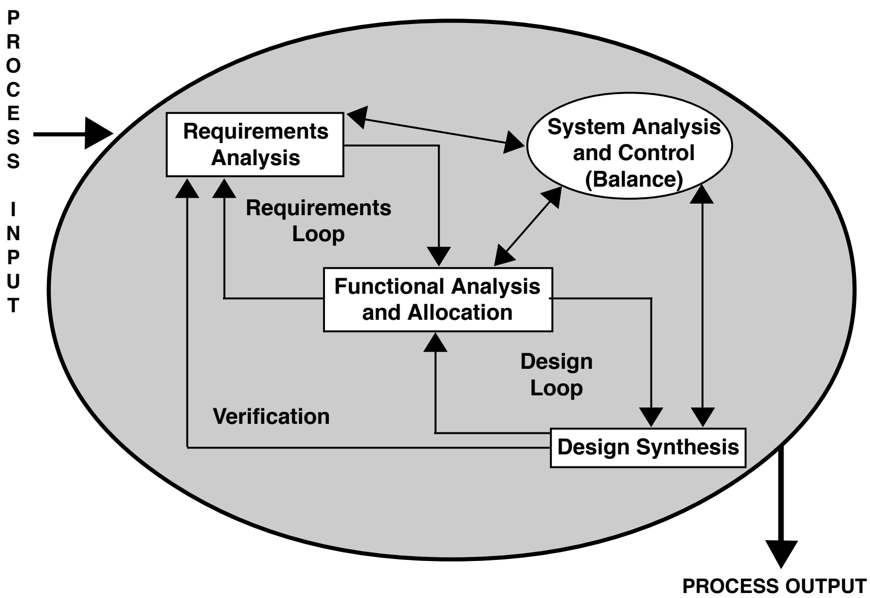


Figure 5.1 - The overall view of the systems engineering process (Anon, 2001)

5.3 Requirements analysis

Approaches for analysis of requirements for complex systems is a much researched topic; for this research, the method recommended in IEEE P1220 standard has been chosen as it is a mature approach that has been used previously by researchers and the industry (Hecht, 1999). The following tasks were thus undertaken to analyse the requirements for the proposed Cross-Technology CNC interoperability system (XTSys):

1) *Identify expectations:* The aim of the research, as identified in chapter 2, indicates that the main expectation is for the cross technology interoperability system to be able to convert machine executable code written for a CNC machine (which is known to be valid) to code that is valid and executable on a machine with a different machining technology with minimal manual intervention.

2) *Define constraints:* As identified in chapter 4, there are many technologies that can be used to manufacturing asymmetric rotational parts with prismatic features. In order to allow the proposed system to be verified, it was decided that XTSys should function with CNC machining technologies that were available to the author.

3) *Identify external constraints:* For the purposes of this research, a 4-axis vertical machining centre and a mill-turn machine were available. These machines were thus identified as the external constraint for the research.

4) *Identify operational scenarios:* Considering that the aim is to enable bi-directional information transfer, the operational scenario for XTSys would be to allow programmes for the machining centre to be converted to programmes for the turn-mill and vice versa. Considering the similarity of the conversion process in both directions, it was decided that scenarios for

conversion from the machining centre to the turn-mill machine would be covered. The same design process can be applied to the reverse conversion and thus following the process once more would have produced no novelty. The operational scenarios are defined based on the example parts shown in Figure 3.32.

5) *Define the measures of effectiveness:* In order to consider the system effective, XTSys should be able to produce a programme for the destination machine that contains no errors and produces a part that is geometrically identical to the part produced on the original machine within the specified tolerances.

6) *Identify system boundaries:* XTSys would only interact with the programmes written for the machine tools as the physical device control and physical connectivity are outside the scope of the research (see section 2.5)

7) *Define the interfaces:* Considering the rich data model offered by ISO14649 (STEP-NC), text files representing populations complying with this standard have been chosen as the main interfaces of the system. Researchers have already addressed the challenges in the conversion of G-Codes to STEP-NC and vice versa (see section 4.5).

8) *Identify utilisation environments:* XTSys should be usable on a normal desktop computer, as it is assumed that these would be easy to procure in all manufacturing environments. It is assumed that it would be desirable for the system to work on the widest variety of hardware and thus Java has been chosen as the programming language to implement due to its portability.

9) *Identify lifecycle process concepts:* analysing the output of the first 8 tasks highlighted no additional concerns in term of lifecycle process concepts.

XTSys would integrate with the existing manufacturing software applications.

10) *Identify the functional requirements:* XTSys should be able to take an ISO10303-21 compliant text file containing an ISO14649 population representing the manufacturing process for a vertical machining centre and produce an ISO10303-21 compliant text file containing an ISO14649 representation of the manufacturing process for a mill turn machine. The components that would be manufactured as the result of running these programmes on their respective machines should be identical.

11) *Identify the performance requirements:* considering that XTSys would not be used in real time, high performance is not a major concern of the research.

12) *Define the modes of operation:* As identified in task 4, the system will be operated in a single mode to convert milling programmes to turn-mill programmes.

13) *Identify technical performance measures:* In the absence of restrictive performance requirements, there are no specific technical performance measures identified for XTSys.

14) *Identify physical characteristics:* XTSys will be realised as a software application and thus no requirements are identified in terms of physical characteristics.

15) *Identify human factors considerations:* Since XTSys is implemented as software, it is assumed that considerations for human factors are already addressed in the design of the hardware and the operating system on which the system will function.

5.4 Functional overview of XTSys

The functional view of the usage of the XTSys for the conversion of one STEP-NC code to another is depicted as an IDEF0 diagram in Figure 5.2. This figure illustrates how to transfer G&M codes from a milling machine to a turn-mill machine by using XTSys, together with the appropriate convertors as the mechanism and manufacturing resources information together with manufacturing standards to regulate the process. These two controls represent the manufacturing dictionary and are described in section 5.4.1.

By converting all NC Codes to STEP-NC, this universal format will be used for both the input and the output data for XTSys. The input can be either directly from a CNC machine, which supports the STEP-NC code format or through a translation of G&M code to STEP-NC code using a conversion system (e.g. UPCi proposed by Zhang (Zhang et al., 2012)).

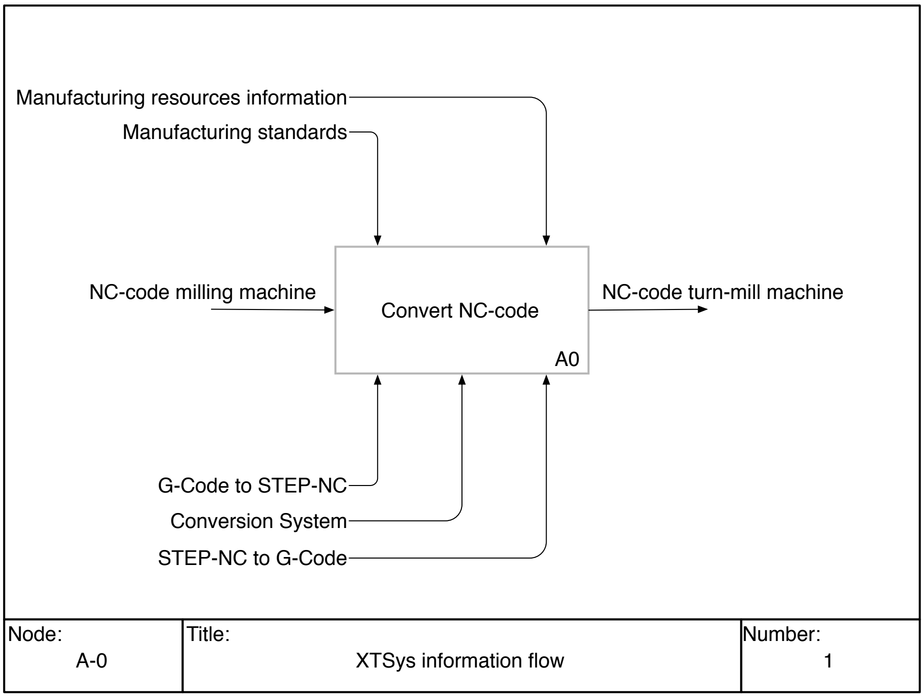


Figure 5.2: IDEF0 view of XTSys information flow

The XTSys output is a STEP-NC file that can be used either with CNC machines which use STEP-NC files as input or can be converted to G&M codes by using a conversion system (e.g. iNet proposed by Nassehi (Nassehi, 2007)).

Figure 5.3 illustrates the information flow between these three systems. All three systems require manufacturing resources information and manufacturing standards as controllers in order to work properly.

The remainder of this section identifies the controls and mechanisms that enable the functionalities of the system together with a decomposition of the conversion function.

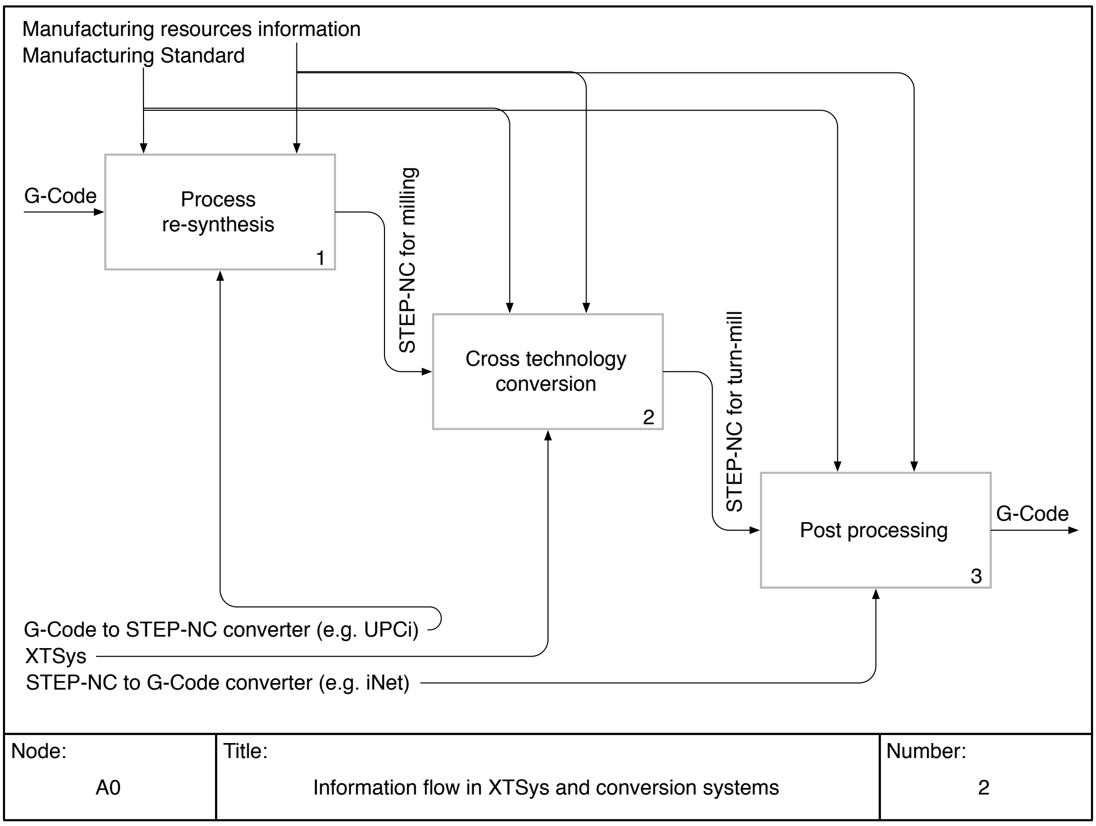


Figure 5.3: Information flow between XTSys and conversion systems

5.4.1 Manufacturing dictionary (Control)

The manufacturing dictionary is a highly flexible database, which is a standardised source (by using latest standards for manufacturing) from which the adapter can get the information for reading, analysing and writing. The manufacturing dictionary should contain four types of information:

i) Machine reference information including the kinematic structure of the machine and its supporting data, which is ISO 14649 part 201. This standard specifies the process data model for manufacturing and machine characteristics for milling, turning and multi-tasking machines (ISO 14649-201, 2011).

ii) Feature meta-data that provides a template for all manufacturing features and their associated operations; this is based on ISO 14649 part 10. This standard specifies the general process data for all NC-programmes which are used in machining technology; it includes geometric and technological information such as definition of the workpiece, a features catalogue that contains all the features information and the basis for operations (ISO 14649-10, 2002).

iii) Operation meta-data that describes various manufacturing operations and their effects based on ISO 14649 parts 11, 12.

iv) Cutting tool reference information including assemblies, which are ISO 14649 parts 111 and 121.

5.4.2 Manufacturing process database (Control)

The manufacturing process database contains the high level manufacturing process data as refined by the analyser, as well as basic manufacturing data such as the cutting tool selection that is used in the CNC machining program that is read by XTSys. The high level information is comprised of manufacturing feature information with the details about the manufacturing operations linked to each feature.

The manufacturing process database, which is an online database, is formed from necessary manufacturing information which is ready for use by the adapter to generate new codes for the new CNC machine. This database uses a java encoding of ISO 14649 in form of the integrated platform for process planning and control (IP^{3}AC) to exchange data with STEP-NC data sources.

IP^{3}AC allow the developers to use JAVA to exchange data between java based object-oriented manufacturing programmes with STEP-NC data sources (Nassehi et al., 2007a). IP^{3}AC with java friendly characteristics and comprehensive access interfaces is an easy tool for reading and writing STEP-NC in this research. Figure 5.4 illustrates classes and interfaces in IP^{3}AC.

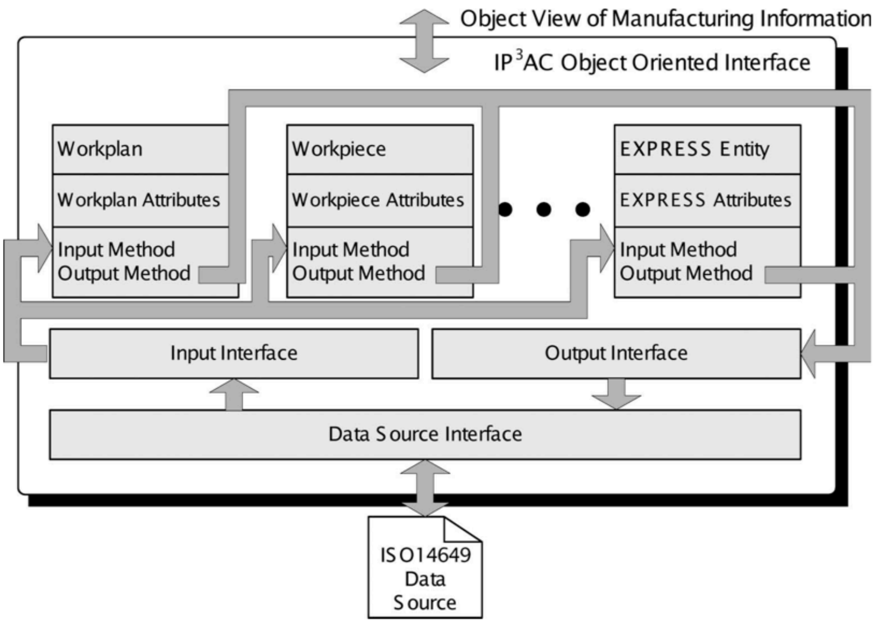


Figure 5.4: IP^{3}AC Classes and interfaces (Nassehi et al., 2007a)

5.4.3 XTSys adapter (Mechanism)

The XTSys adapter is comprised of three components or sub-mechanisms that divide the expected functionality of the system into three distinct areas: a reader, analyser and writer. The XTSys reader identifies the features, operations and cutting tools information. This information will be used in the cross technology converter which determines new corresponding features and operations for the destination machine by using the XTSys analyser mechanism. The features and operations for the new machine will then be used to generate output as a new STEP-NC file through the XTSys writer mechanism. The overall flow of information in XTSys is shown in Figure 5.5.

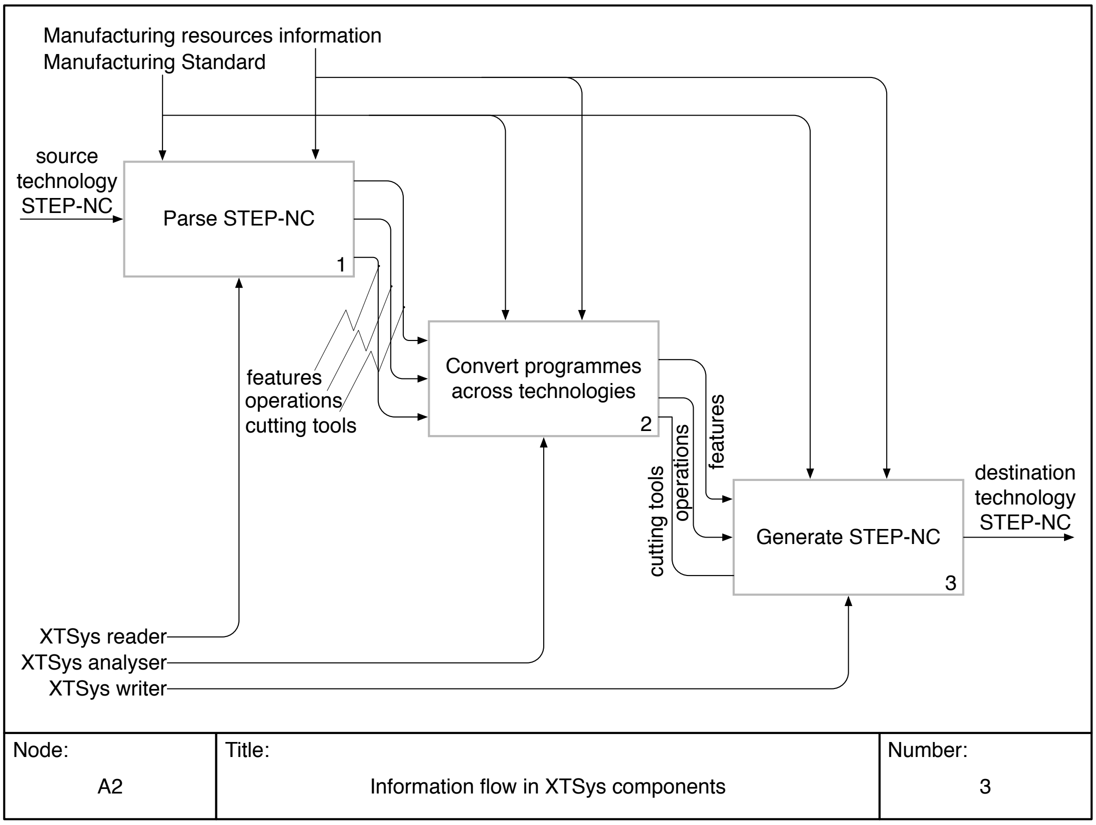


Figure 5.5: Information flow in XTSys components

5.4.3.1 XTSys reader

The reader parses through STEP-NC from source and utilises the manufacturing dictionary to identify items of information pertaining to the various aspects of the process, such as workpiece, manufacturing features, machining operations, working steps, strategies, etc. before storing this manufacturing process data in the XTSys manufacturing database. As the manufacturing dictionary is extendable, with the addition of the necessary meta-data, it is possible to adapt XTSys to read machine code written in any language (e.g., G&M codes, STEP-NC, Heidenhain language, etc.) for input and output.

Figure 5.6 provides an overview of the information flow in the reader. STEP-NC codes are input into the system, machine working steps are identified in STEP-NC code and from this information manufacturing features and machining operations are determined.

Tools and technology information are then determined from the machining operations and the feature geometry is determined from the manufacturing features. After identifying the information for the adapter, the reader will store all the information into the manufacturing process database for the analyser to analyse and generate the necessary information for the destination machine. Figure 5.7 illustrates the information flow in the reader.

Feature identification, which is done by the XTSys reader, will identify the feature type (e.g. `step`, `pocket`, `slot`, `planar_face` and `profile_feature`) and then recognise each feature to identify the necessary attributes. For example, if the reader recognises a pocket, it will identify the pocket type as either an open or a closed pocket.

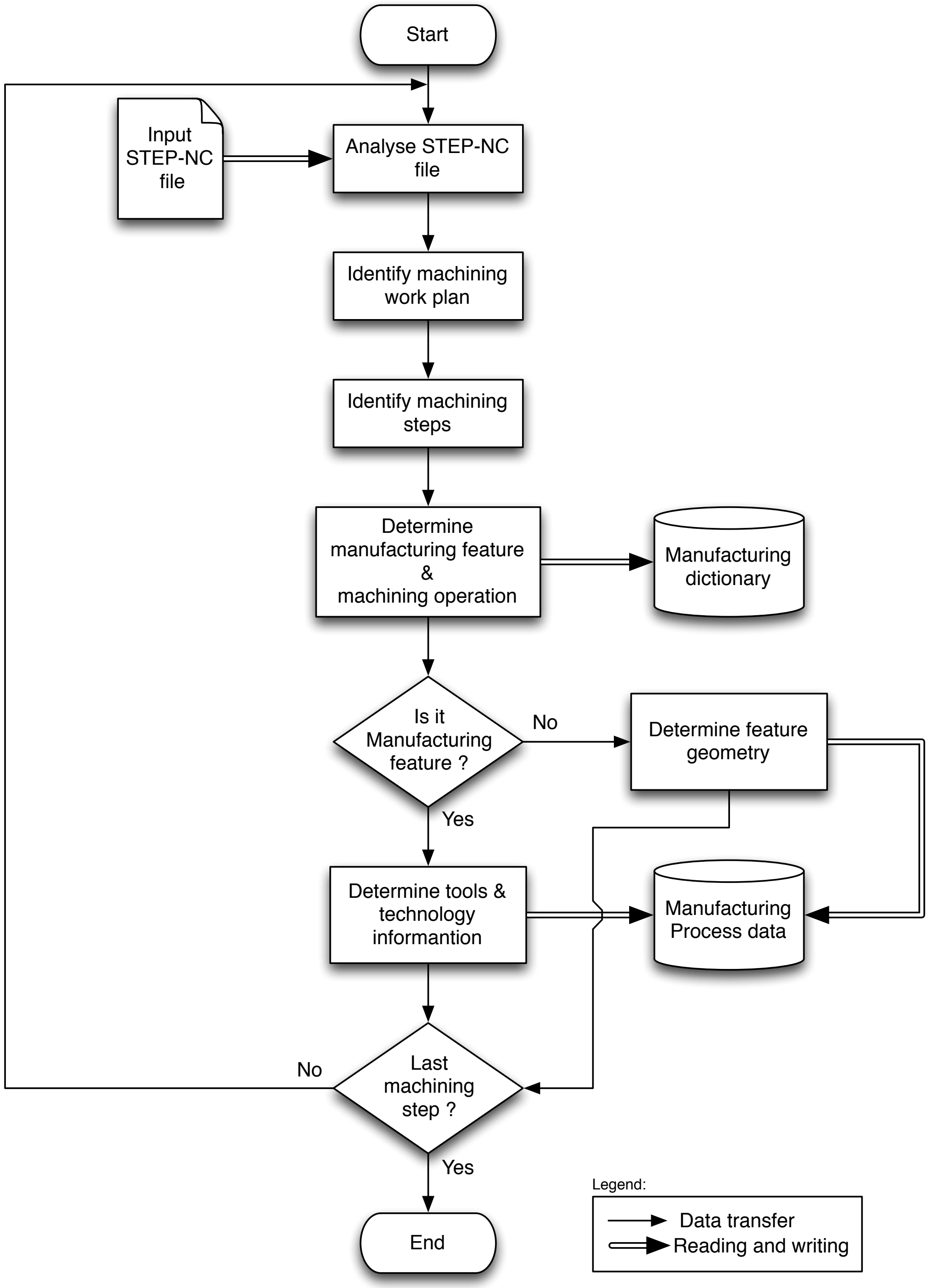


Figure 5.6: Reader process flowchart for STEP-NC code

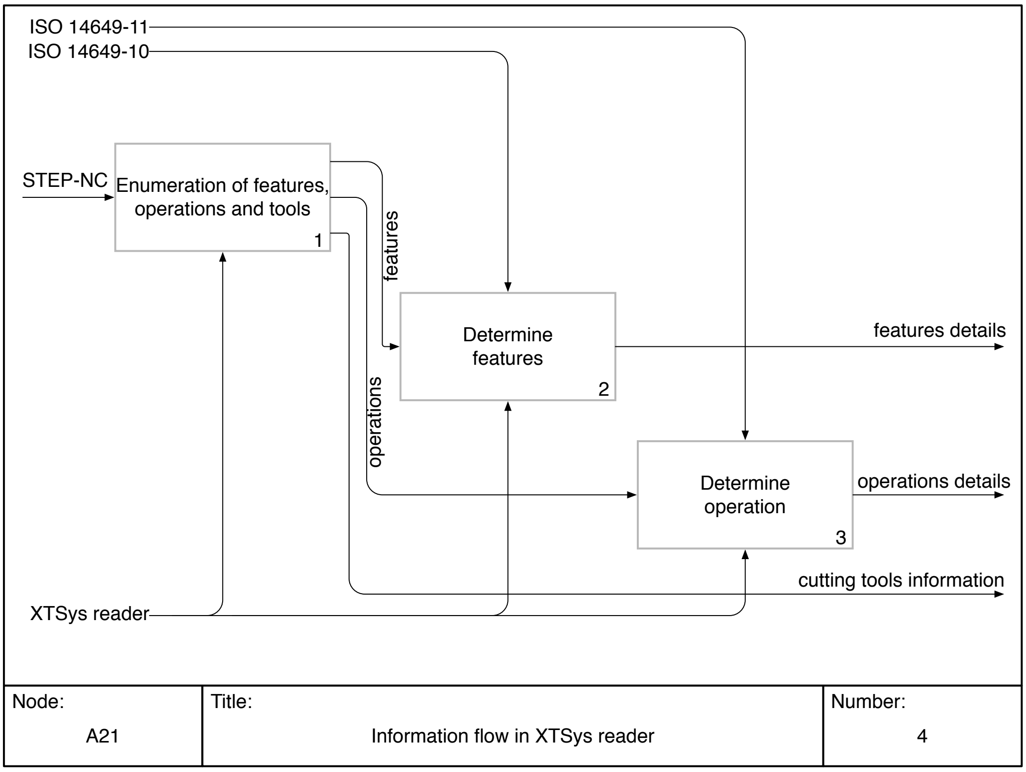


Figure 5.7: Information flow in XTSys reader

If the pocket is a `closed_pocket` then the `feature_boundaries` will be identified. And if the pocket is an `open_pocket` then the reader will identify `open_boundarys`. The information in this stage will be used by the XTSys writer and analyser to generate new a STEP-NC file for the new CNC machine.

The XTSys reader using IP^{3}AC and ISO 14649 part 11, identifies the type of operations as either a milling type or a drilling type. If it is a milling type then the type of milling operation such as `plane_milling`, `side_milling` or `bottom_and_side_milling` will be identified, and if it is a drilling type then it will identify whether it is `drilling`, `boring`, `back_boring`, `tapping` or `thread_drilling`. This information will be used by the XTSys analyser to identify new operations for the new CNC machine.

5.4.3.2 XTSys analyser

The second component of the XTSys adapter is the semantic analyser which interprets the manufacturing process data and finds semantically equivalent constructs for generating new STEP-NC codes that are technologically compatible with the destination machine. These are stored in the XTSys manufacturing database and linked to the original data items.

After gathering the information from the source code by the reader, the XTSys analyser's role is to convert machining operations and machining features information from source to destination based on the availability of tools and operations in the destination machine. To realise such a system, the analyser should categorise the machining operations and features to different sublevels and then start to find the operations and features from the manufacturing process database that are suitable for the destination CNC machine based on the machine information.

These transformations should only be made if they are semantically equivalent. The analyser uses a set of predefined rules to assess the equivalence of various operations in each case based on the type of features. Figure 5.8 shows a basic instance of these rule sets; the rule sets are entitled ''semantic transformation templates'' to denote the fact that replacement of operations does not modify the results achieved by running the programme and, in essence, the operations carried out on their respective machines yield the same result. In the shown semantic transformation template, the `facing` operation in the turning context is the equivalent of `plane_milling` in the milling context and, similarly, the `bottom_and_side_milling` or `side_milling` operations are the equivalent of `contouring` in the turning context.

The main roll of analyser is to find a right feature for destination machine, in this research six common feature were chosen as the main focus: planar face, profile feature, round hole, slot, step and pocket.

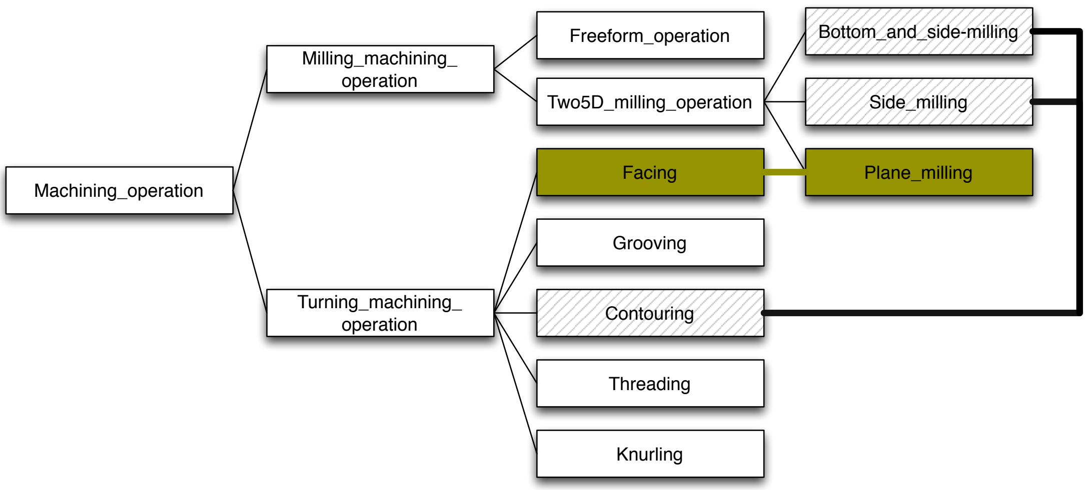


Figure 5.8: Basic semantic transformation template in the XTSys analyser for cylindrical rotational parts

After the semantic transformation, another rule set in the XTSys analyser is activated to determine the choice of operations based on availability of cutting tools and the method of setup of the workpiece in the machine. Figure 5.9 illustrates an excerpt of this rule set in the XTSys analyser. As illustrated in Figure 5.9, the XTSys analyser reads the destination machine information from the manufacturing dictionary and then, by considering each manufacturing feature, assesses the best manufacturing option based on the capability of the destination machine for performing various operations.

For example, for manufacturing a pocket on a turning centre, if the destination machine has milling capability in the appropriate axes, the XTSys analyser will simply use the milling operation, whereas if the destination machine does not have this capability, then the analyser will check the pocket to assess whether it is machineable using an alternative approach. In the given scenario, there is no viable machining method for an open pocket but for a closed pocket the XTSys analyser will check to assess if the pocket is located at the centre of the workpiece. Should the pocket have a circular boundary and be located at the centre of the workpiece, the analyser will choose the appropriate turning operation to manufacture the pocket. It is also conceivable that a non-centric pocket could be manufactured using special non-concentric tooling and therefore the availability of such resources is also checked in the XTSys analyser.

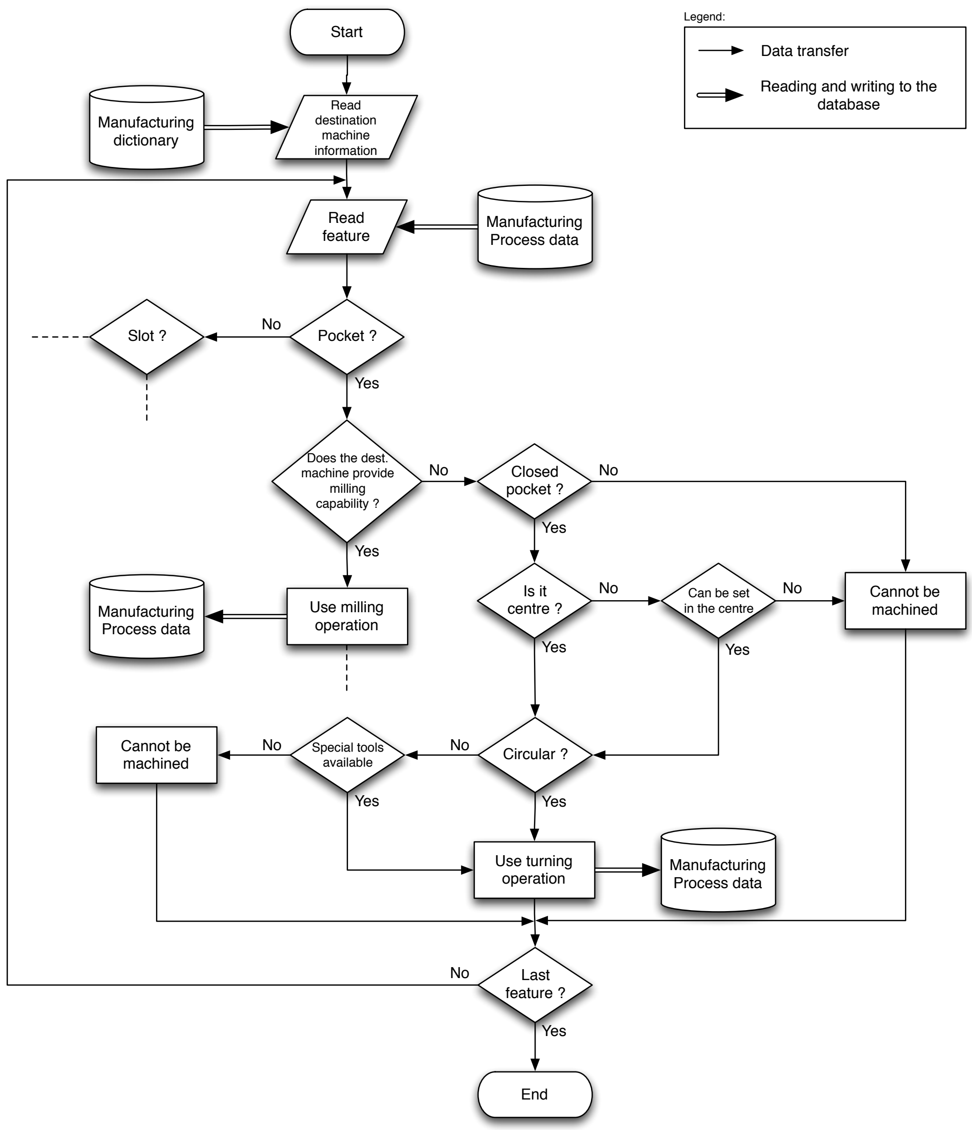


Figure 5.9: An excerpt of operation determination rule set in the XTSys Analyser.

Figure 5.10 illustrates the information flow in the XTSys analyser: operations, features and cutting tools information are inputs for XTSys analyser from the XTSys reader. New features and operations are the output of XTSys analyser after conversion of features and operations by semantic analyser.

Feature details which are inputs for the feature converter serve to distinguish the milling features and the turning features from each other. The feature selector will then select the right feature for the destination machine by using ISO 14649 part 10 and 12 together (the step by step semantic transformation of features described in section 6.3).

Features information flow in XTSys is illustrated in Figure 5.11.

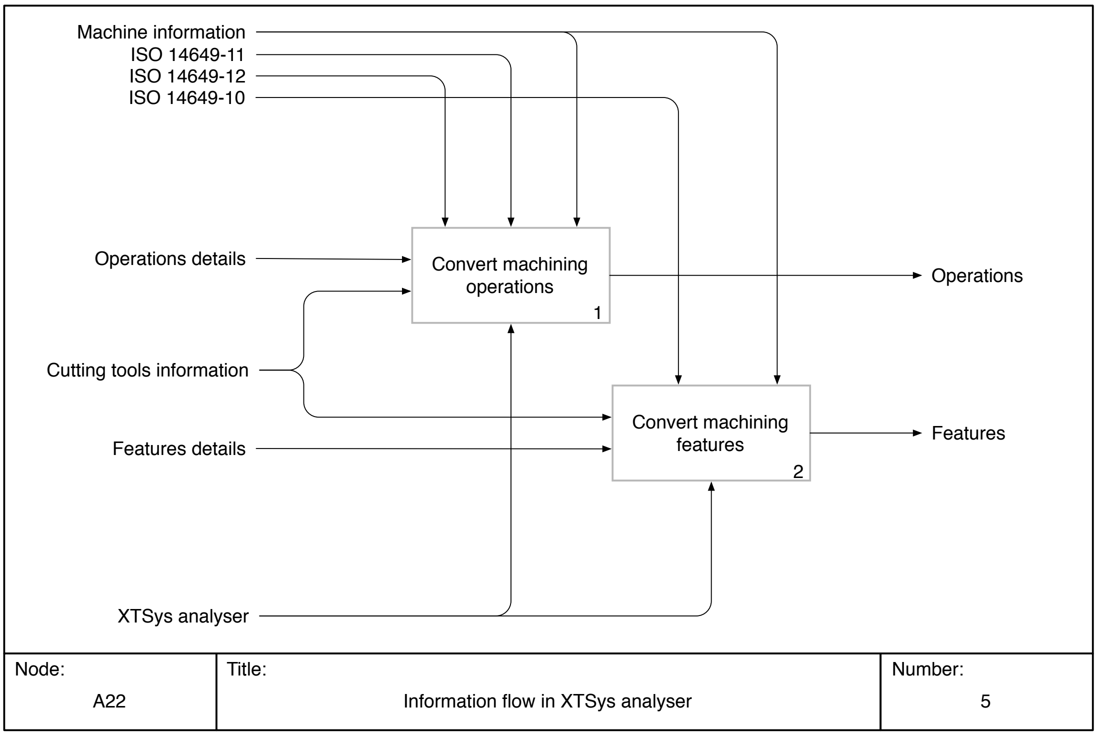


Figure 5.10: Information flow in XTSys analyser

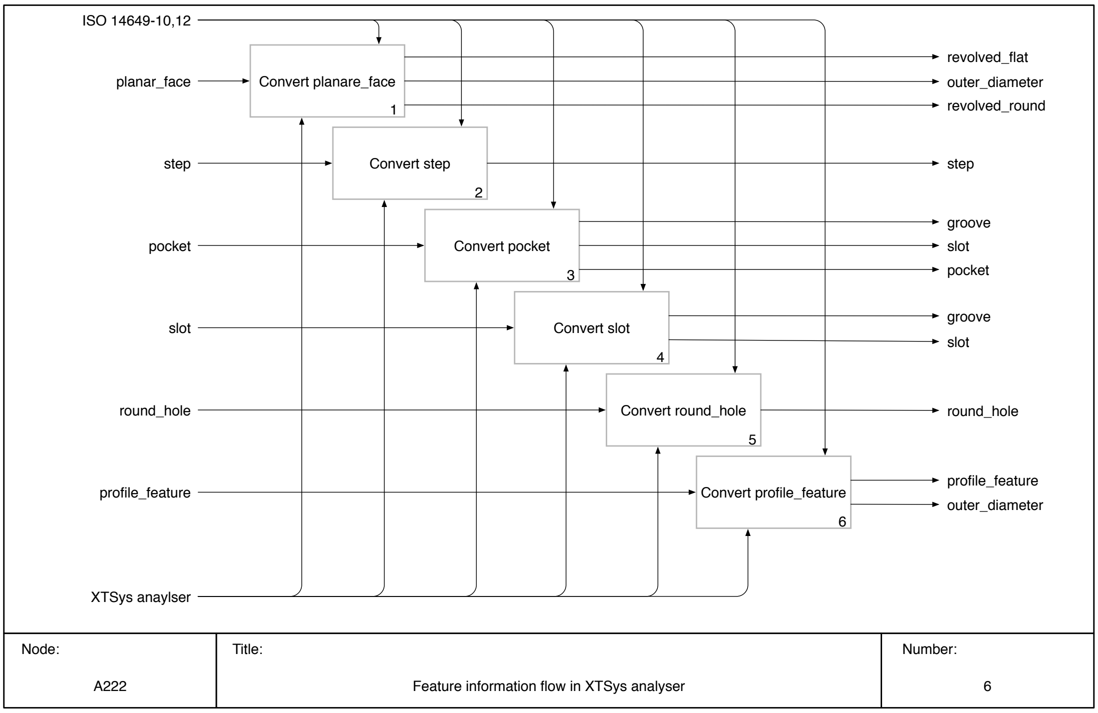


Figure 5.11: Feature information flow in XTSys analyser

The operation details, which are milling operations, are analysed by the machine operation selector to become either turning operations or milling operations. In this process the machining operations selector uses ISO 14649 parts 11-12 together to choose the correct operation for the destination feature (the step by step operation conversion is described in 6.3). Figure 5.12 illustrates the information flow in the operation converter of XTSys analyser.

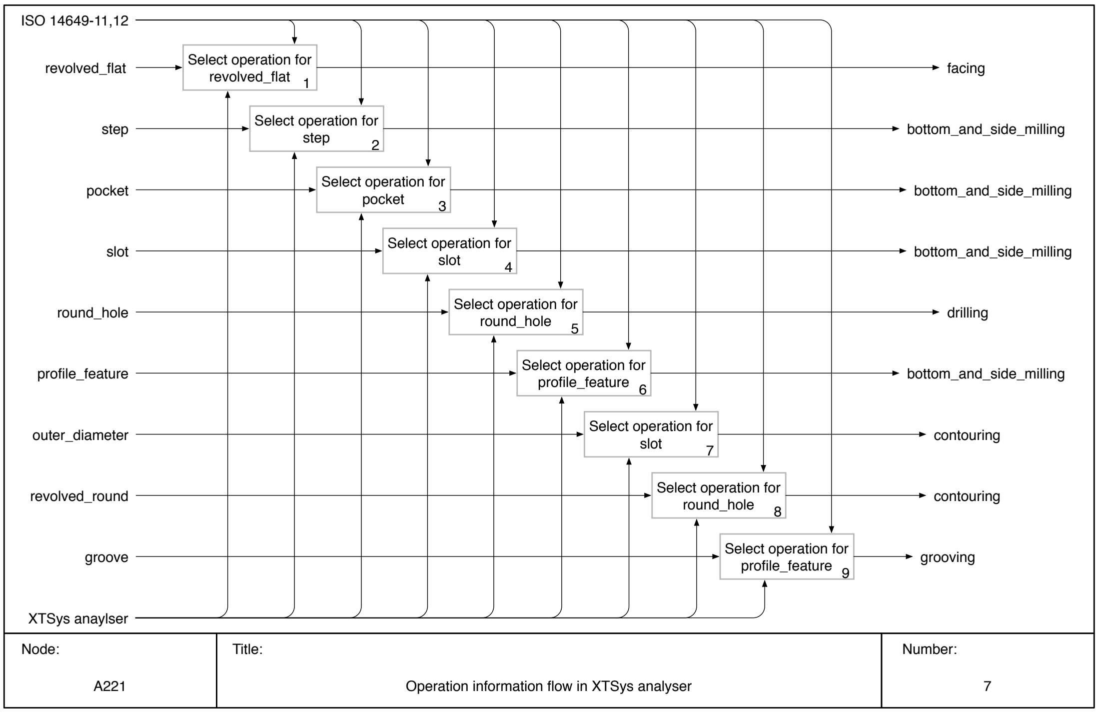


Figure 5.12: Operation information flow in XTSys analyser

5.4.3.3 XTSys writer

The third XTSys adapter component, the writer, combines this generated information with the machine information from the manufacturing dictionary to generate an appropriate NC code for the destination machine.

As illustrated in Figure 5.13, the writer starts to read the destination machine information from the Manufacturing Dictionary to generate the Header and Data section of the STEP-NC file. The information for the header is similar to the information for the source machine. For the data section the writer reads the first operation from the manufacturing process database, (which is carried through from the analyser decision) and then reads the features and operations (which is chosen for the current technology). After reading the operation and the feature, the writer generates the code based on information in the manufacturing dictionary and writes the STEP-NC code to a file. The writer finishes the process at the last operation.

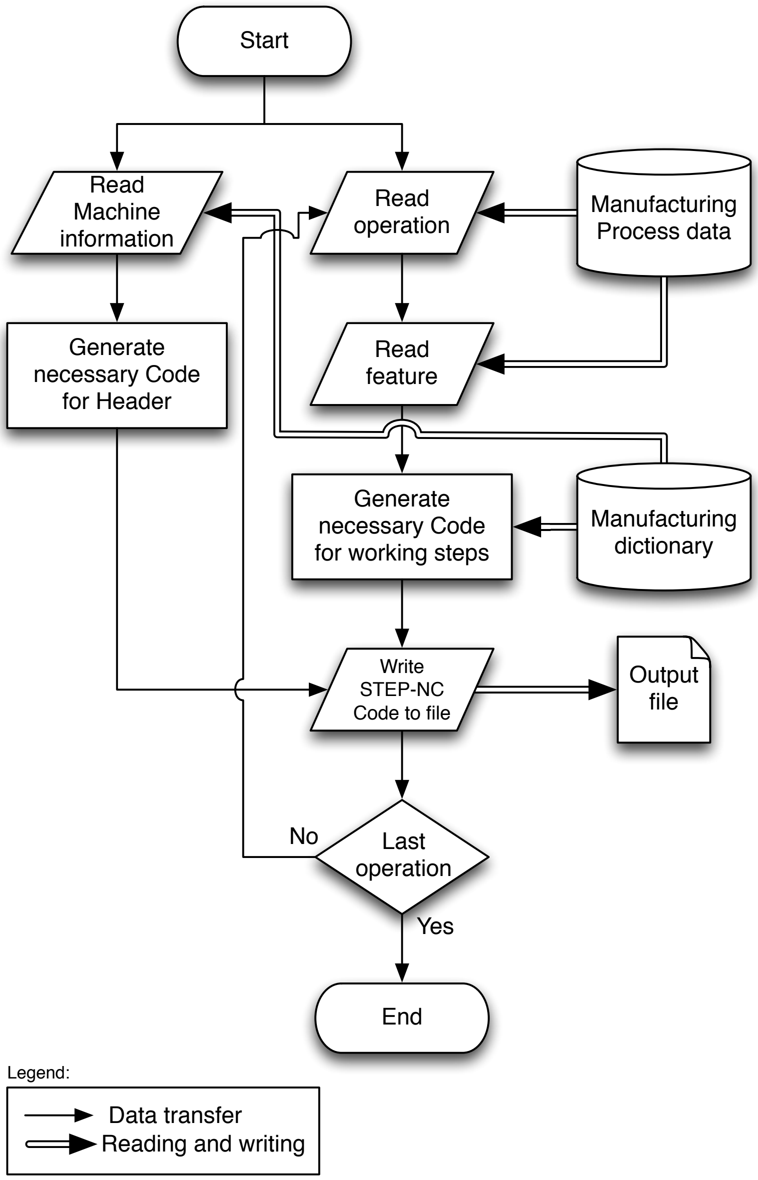


Figure 5.13: Writer process flowchart

Figure 5.14 illustrates the information flow in the writer. The writer gets the turning and milling operations from the analyser output as an input and synthesises them to generate new code for the destination machine. The writer then uses workpiece and tools information, which it will use in the destination machine to generate the new code. The code is then written by XTSys writer and IP^{3}AC.

5.5 Visualisation of information flow through XTSys

In order to establish the requirement loop of the systems engineering paradigm, the example part in Figure 5.15 comprising of two basic features has been chosen to check the flow of information in the analyser. Following the activities modelled in section 5.4 should convert the milling programme for the part to one that is suitable for a turn-mill machine.

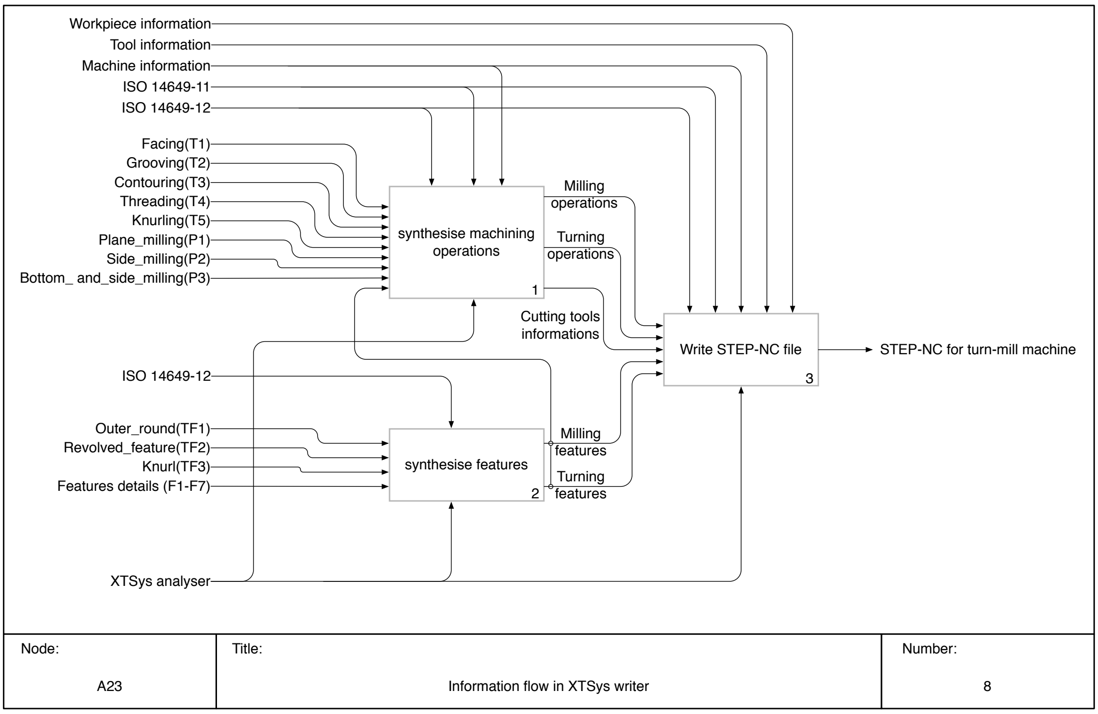


Figure 5.14: Information flow in XTSys writer

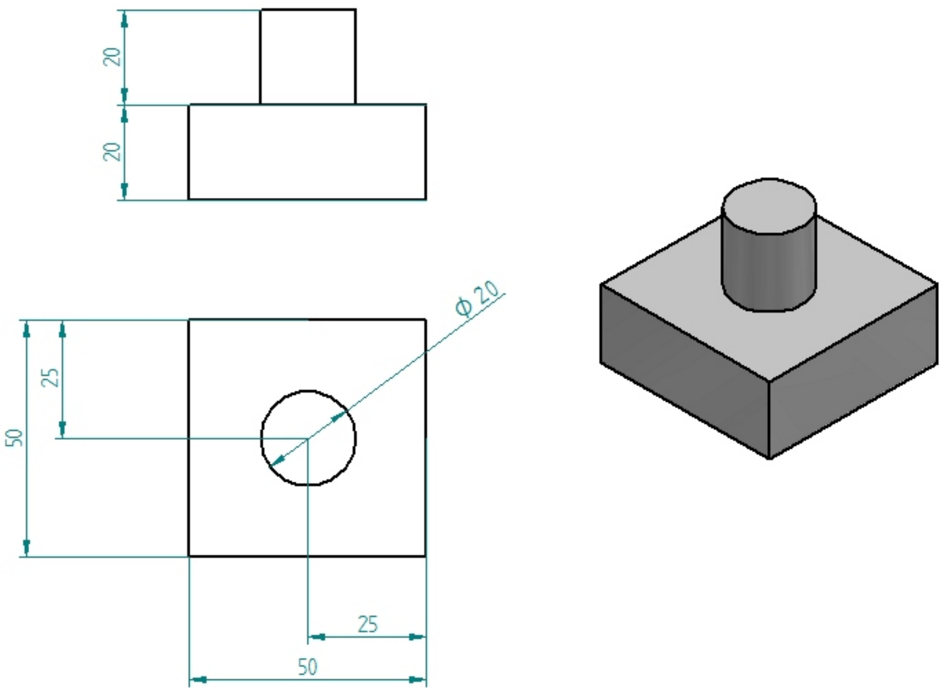


Figure 5.15: Example part

For machining this part with a milling machine, two operations are required:

1. `Plane_milling`: to create a smooth surface on the topside of the part.
2. `Bottom_and_side_milling`: to create the cylindrical boss.

The steps have to be carried out in a certain order; plane milling operation should be finished before the bottom and side milling operation. It is expected that the program generated by the system for the turning will retain the same order.

Figure 5.16 illustrates the features and operations for machining the part with milling operation.

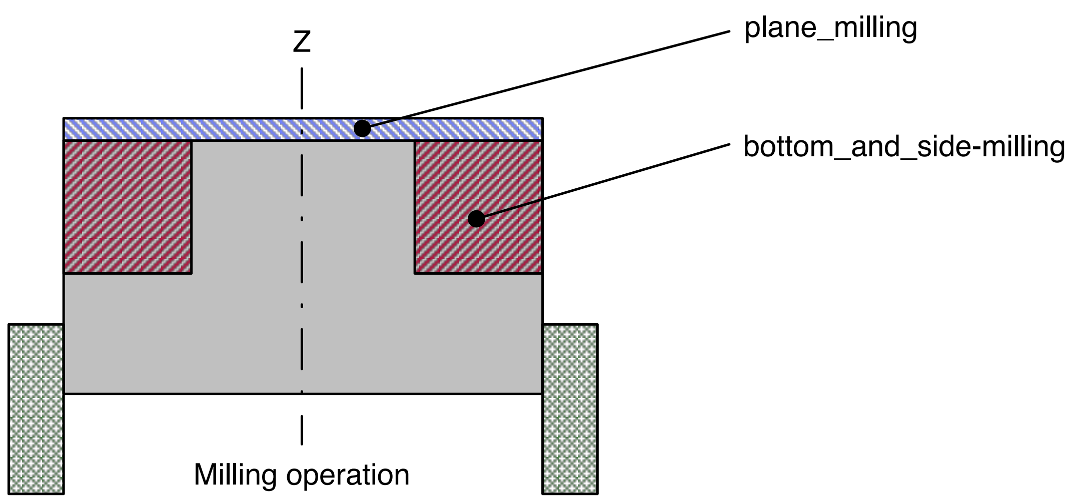


Figure 5.16: Milling operations for example part

Excerpts of the STEP-NC program that containing the milling manufacturing process is shown in Table 5.1.

Table 5.1: An excerpt of the STEP-NC milling program written to machine the example component

```
#4=WORKPIECE('MAIN WORKPIECE',$,$,$,$,$,());
#21=PLANAR_FACE('FACE1',#4,(#50),#51,#52,#53,#54,$,());
#50=PLANE_FINISH_MILLING($,$,'FINISH PLANAR
FACE1',$,$,#70,#71,#72,$,$,$,$,$,$);
#51=AXIS2_PLACEMENT_3D('FACE1',#103,#104,#105);
#52=ELEMENTARY_SURFACE('FACE1-DEPTH',#106);
#53=LINEAR_PATH($,#110,#111);
#54=LINEAR_PROFILE($,#113);
#70=MILLING_CUTTING_TOOL('MILL 66MM',#100,(#101),$,$,$);
#71=MILLING_TECHNOLOGY($,.TCP.,$,$,$,.F.,.F.,.F.,$);
#72=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.F.,$,(),.F.,$,$,());
```

This program was taken through the activities to generate a turning program to machine the same part. To generate the new code, XTSys reads the code to find the first feature with is `planar_face` and the operation for this is `plane_milling`. According to the destination machine, which is turning, `planar_face` is similar to `revolved_flat` (this is the same feature for facing in turning technology), XTSys adapter is changing the feature to turning one and according to operation available in destination machine, `end_face` is chosen for manufacturing the `revolved_flat` feature. The same scenario for the second feature was flowed by XTSys adapter to generate the new code. Figure 5.17 illustrates the new operations for turning machine.

Table 5.2 shows excerpts from the resulting STEP-NC turning program. As seen in the input and the output code, XTSys would change the `Plane_finish_milling` in line 50 in the milling program to `Facing_finish` in the same line in the turning program. The specification of the face remains consistent and the order of operations is preserved. As such the semantics of both programs are equivalent and thus the functional model of the system is verified against the requirements.

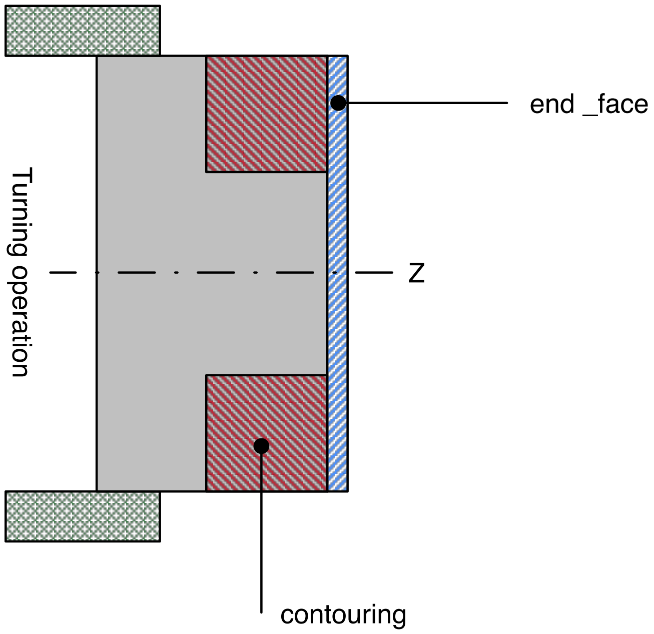


Figure 5.17: Turning operations for example part

Table 5.2: An excerpt of the generated STEP-NC turning program for machining the example component

```
#4=WORKPIECE('MAIN WORKPIECE',$,$,$,$,$,());
#21=REVOLVED_FLAT('FACE1',#4,(#50),#51,#52,#53,#54,$,());
#50=FACING_FINISH($,$,'FINISH END FACE',$,$,#70,#71,#72,$,$,$,$);
#52=ELEMENTARY_SURFACE('FACE1-DEPTH',#106);
#53=LINEAR_PATH($,#110,#111);
#54=LINEAR_PROFILE($,#113);
#70=TURNING_MACHINE_TOOL('FINISHING TOOL',#111,(#112),$,$,$);
#71=TURNING_TECHNOLOGY($,.TCP.,$,$,.F.,.F.,.F.,$);
#72=TURNING_MACHINE_FUNCTIONS(.T.,$,$,(),.F.,$,$,(),$,$,$);
```

6 Development of a prototype of the XTSys

6.1 Introduction

In order to demonstrate the effectiveness of the XTSys framework in accordance with the systems engineering paradigm, a prototype implementation has been developed using Java.

In addition to the identified requirement for portability (see section 5.2), Java was chosen due to compatibility with the existing G-Code to STEP-NC and STEP-NC to G-Code convertors (UPCi and iNet) and the availability of a single library for manipulating STEP-NC manufacturing information in an object-oriented manner (iP^{3}AC).

This chapter presents the structure of the prototype, the semantic transformation functions implemented for converting milling programmes to turn-mill programmes and an overview of the user interface of the prototype.

6.2 Structural overview of the XTSys prototype

The structural overview of XTSys is shown in figure 6.1. The machining programme is read by the system as STEP-NC data structures by the reader component of the XTSys adapter.

The information is then translated into standardised syntax and passed on to the analyser where it is refined to high level manufacturing process data. When required, the data is passed on to the writer component in the XTSys adapter to be fused with the resource information for the destination machine obtained from the manufacturing dictionary. Figure 6.2 illustrates a class diagram for the components in XTSys.

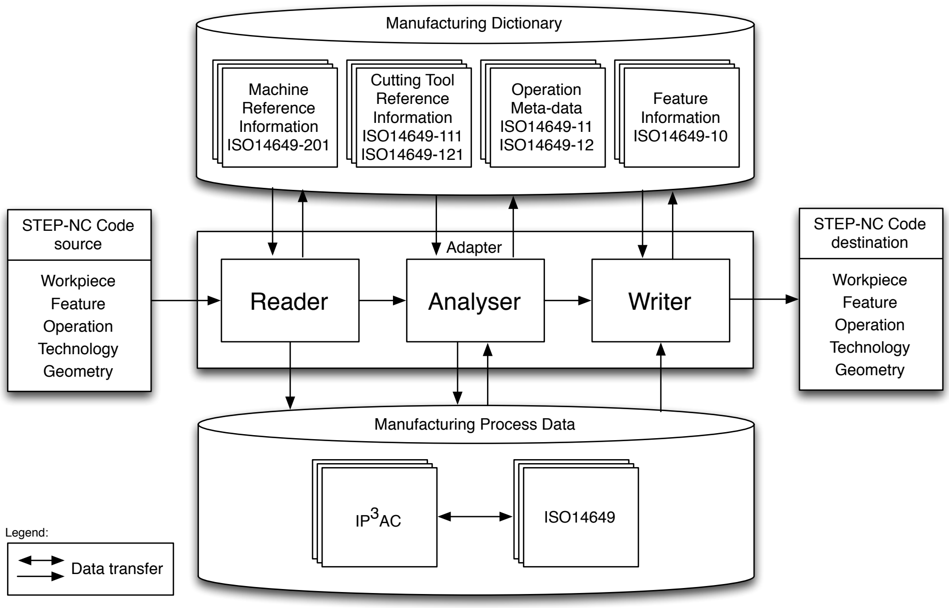


Figure 6.1: The structure of the Cross-Technology CNC interoperability system

6.3 Semantic transformation templates

6.3.1 Planar face

Planar face is one of the machining features in ISO 14649 part 10, which is used to describe the machining of straight faces on the workpiece..

Usually, in milling, this feature is used for facing operations at the beginning of the machining process so as to have a clean surface on the block. In turning, according to planar face elementary surface from the milling code, a different approach needs to be adopted. For example, planar face in milling technology will use a plane milling operation, whereas after conversion to turning technology, the feature will be revolved flat and operation will be facing. Figure 6.3 shows the operation in both technologies.

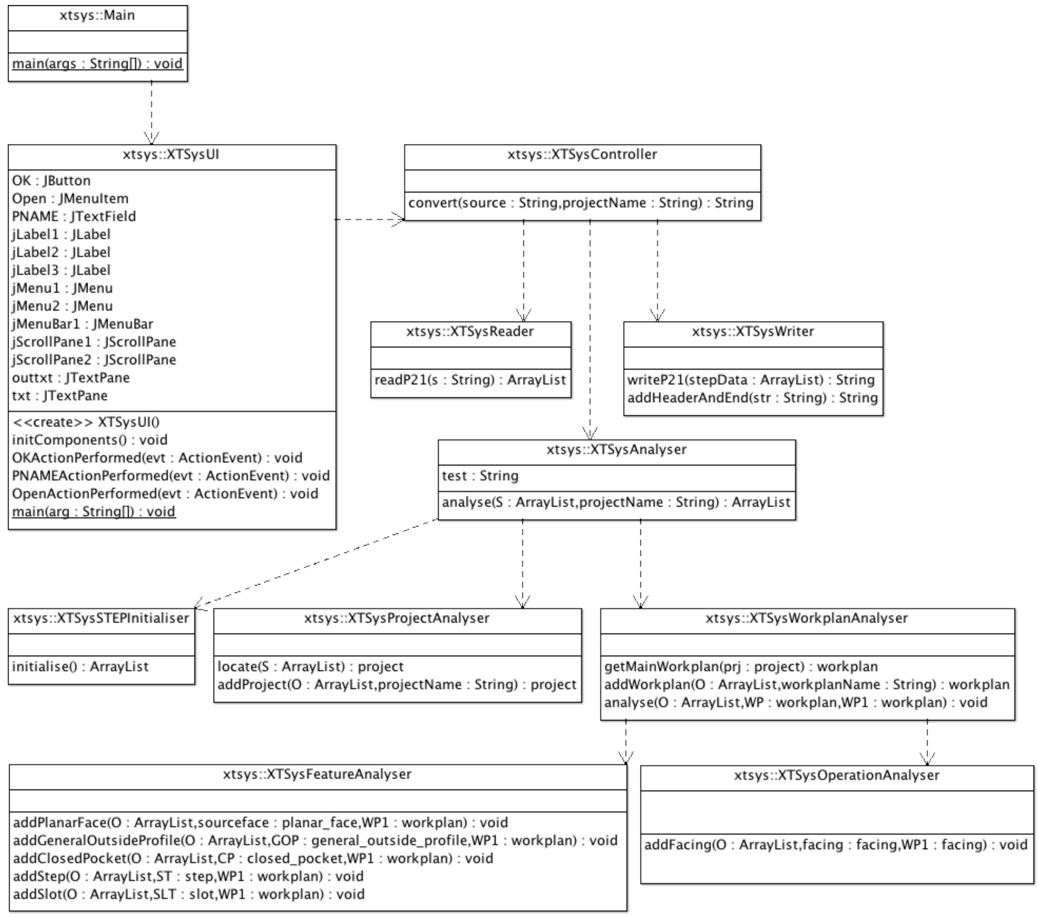


Figure 6.2: Class diagram of the XTSys prototype

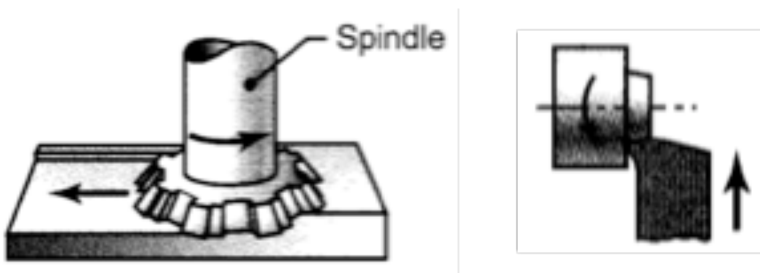


Figure 6.3: Plane milling operation in milling (left) and facing operation in turning (right) adapted from (Kalpakjian, 2014)

According to ISO 14649 part 10, `planar_face` has a number of major elements to be identified, these are: `depth`, `course_of_travel`, `removal_boundary`, `face_boundary` and `its_boss`. Figure 6.4 illustrates these elements.

The information in the planar face entity is:

- `Depth`: the depth denotes the lowest point of the material that needs to be taken away from the workpiece to reach the final shape of the feature. Depth is one of the machining feature entities and is described by a plane which includes the lowest points of the feature, if the depth is not an orthogonal plane to the z-axis then, depending on the explicitness of the description of the features, there can be more complex bottom shapes;

- `Course_of_travel`: this shows the distance and direction the tool needs to travel to remove the material,

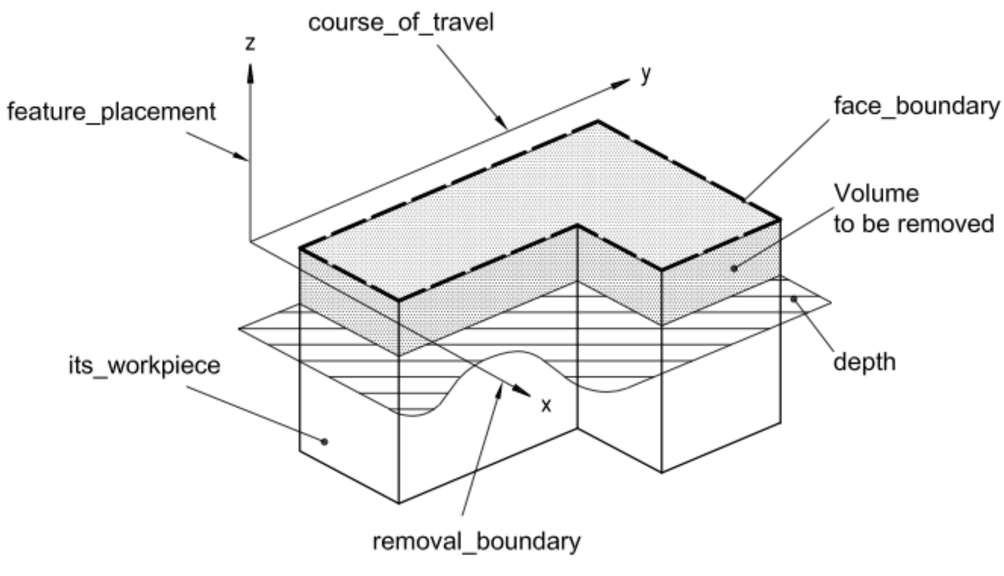


Figure 6.4: Planar face (ISO 14649-10, 2002)

- `Removal_boundary`: this shows the boundary of the material that needs to be removed,

- `Face_boundary`: this is an optional entity to finalise the final shape of the workpiece after the planar cut has been applied,

- `Its_boss`: an optional list of entities. A boss of a planar face defines a part of the face which is not cut during the manufacturing process.

In milling operations, the depth of features is always along the z-axis with the removal boundary along the x-axis and the course of travel in simple facing on the y-axis. For machining this feature in a milling machine, a `plane-milling` operation will be used.

For machining the same operation in a turn-mill machine, the analyser needs to read the `planar_face` information from the milling code and chose the correct feature in the turning machine. The bottom of the `planar_face` is the most critical element among the features that the analyser needs to assign a correct feature for turning.

From the depth entity in `planar_face`, which is `elementary_surface`, there are five different bottom conditions that need to be converted to turning features. These are: plane, `cylindrical_surface`, `conical_surface`, `spherical_surface` and `toroidal_surface`. Each one of these surfaces has specific data which the analyser uses to form a new entity in the corresponding features for turning. These correspondences are listed as follows:

- `Plane` to `revolved_flat`,

- `Cylindrical_surface` to `revolved_flat`,

- `Conical_surface` to `outer_diameter`,

- `Spherical_surface` to `revolved_round`,

- `Toroidal_surface` will remain same.

`Revolved_flat` is a type of revolved feature in ISO 14649 part 12, which carries information such as flat edge shape, material side and radius, as shown in Figure 6.5.

The information in the revolved flat entity is:

- `Flat_edge_shape`: which is a linear profile that when revolved about an axis defines the shape of an area of the part,
- `Material_side`: that specifies the material removal direction,
- `Radius`: the distance from the axis of rotation to define placement of the profile that will be swept about the axis.

For the `planar_face` with plane surface, `course_of_travel` will be the new entity for `flat_edge_shape` in `revolved_flat` and the radius of `revolved_flat` will be equal to zero. If the depth in planar face was more than zero then the location of the feature on the turning machine will be changed.

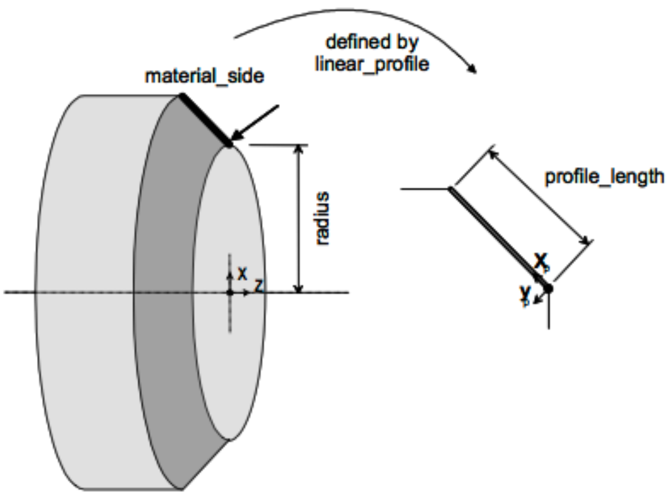


Figure 6.5: Revolved flat (ISO 14649-12, 2004)

```
If the depth of the planar_face = plane
then the manufacturing_feature = Turning feature
      the turning_feature = revolved_flat
      which radius = 0
      and flat_edge_shape = course_of_travel
      and material_side = direction
End if

If depth of machining_feature for planare_face in milling operation
> 0
  then End_face : location = (0.000, 0.000, z-depth)
End if
```

For the `planar_face` with `cylindrical_surface`, the radius in `cylindrical_surface` will be the new entity for `flat_edge_shape` in `revolved_flat` and the radius `of revolved_flat` will be equal to zero. If the depth in planar face was more than zero then the location of the feature on the turning machine will be changed.

```
If the depth of the planar_face = cylindrical_surface
then the manufacturing_feature = Turning feature
      the turning_feature = revolved_flat
      which radius = 0
      and flat_edge_shape = radius of the cylindrical
      and material_side = direction
End if

If depth of machining_feature for planare_face in milling operation
> 0
  then End_face : location = (0.000, 0.000, z-depth)
End if
```

For the `planar_face` with `conical_surface`, the `outer_diameter` feature will be chosen. `Outer_diameter` is one of the subtypes of `outer_round` feature, which carries information such as `diameter_at_placement`, `feature_length` and `reduced_size`. These entities are illustrated in Figure 6.6.

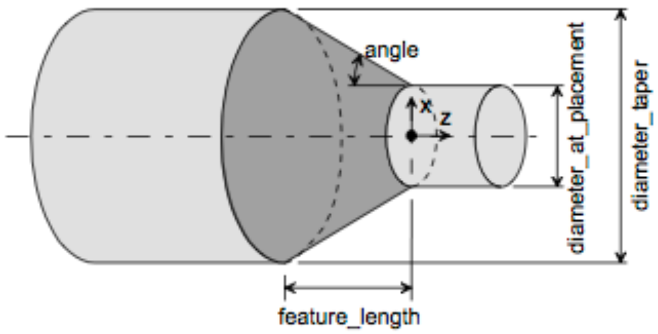


Figure 6.6: Outer diameter with taper adapted from (ISO 14649-12, 2004)

The information in the outer diameter entity is:

- `Diameter_at_placement`: which describes the diameter at the side of the feature, where the origin of the feature is defined,
- `Feature_length`: the length of the feature,
- `Reduced_size`: which defines the cone information in the feature. There are two pieces of information needed to distinguish the cone: `diameter_taper` and `angle_taper`.

In `outer_diameter`, the following data from `conical_surface` in `planar_face` will be the new data for turning.

```
If the depth of the planar_face = conical_surface
Then the manufacturing_feature = turning feature
      the turning_feature = outer_diameter
      which diameter_at_placement = conical_surface radius * 2
      and feature_length = depth
      and reduced_size = one of the taper_select
            which if it is diameter_taper then
                  final_diameter = 2 * ( conical_surface
radius + depth * tg semi_angle )
            else ( it is angle_taper ) then
                  angle = semi_angle
End if
```

For the planar_face with spherical_surface, the revolved_round feature will be chosen. Revolved_round is one of the subtypes of revolved_feature, which carries the same information as revolved_flat, i.e. flat_edge_shape becomes rounded_edge_shape. This feature is shown in Figure 6.7.

The information in the outer diameter entity, therefore, is:

- Rounded_edge_shape: specifies the arc that when revolved about an axis defines the shape of an area of the part.

The following data from spherical_surface in planar_face will be the new data for turning.

```
If the depth of the planar_face = spherical_surface
Then the manufacturing_feature = turning feature
      The turning_feature = revolved_round
      Which rounded_edge_shape = radius from spherical_surface
      and radius = 0
      and material_side = direction
End if
```

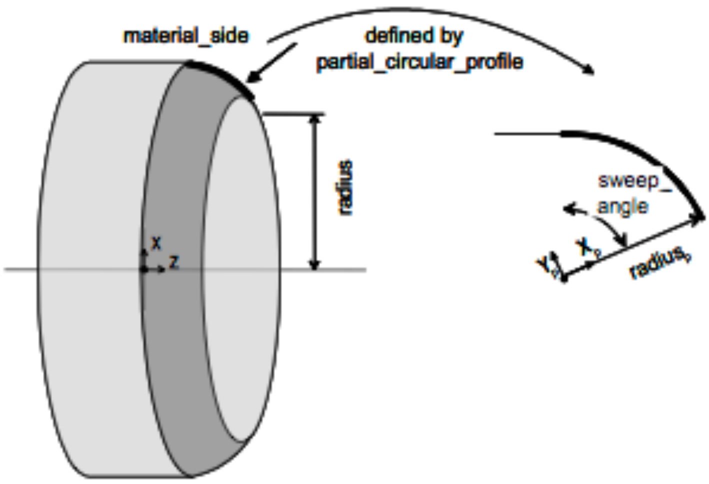


Figure 6.7: Revolved round adapted from (ISO 14649-12, 2004)

`Planar_face` with `toroidal_surface`, it is difficult to manufacture this feature according to lack of machine and tool technology. For this reason this feature will be not included in this research.

```
If the depth of the planar_face = toroidal_surface
then the manufacturing_feature = planar_face with toroidal_surface
```

By identifying features and operations in the source machine, the analyser needs to generate the feature for the destination machine (i.e. a turn-mill machine with milling or turning features) and then choose the right operation for the feature in the turn-mill machine. Operations for the features that the analyser has translated from the milling machine to the turn-mill machine are described below.

```
If the turning feature = revolved_flat
then turning_machining_operation = facing
end if

If the turning feature = outer_diameter
then turning_machining_operation = contouring
end if

If the turning feature = revolved_round
then turning_machining_operation = contouring
end if
```

6.3.2 Profile feature

One of the other features in milling is the profile feature, which is a volume of material removed from the boundary shape of the workpiece. According to the types of profile feature in the milling machine, there will be corresponding features in the turn-mill machine to manufacture the same part. Profile feature has two abstract subtypes, (a) `general_outside_profile` and (b) `shape_profile`.

(a) General_outside_profile

A general outside profile is the removal volume of a material from the outside of the feature boundary. Figure 6.8 illustrates the feature and its entities.

The information in the general outside profile entity is:

- `Feature_boundary`: which is the contour of the profile to be followed by the tool, this entity identifies the profile which can be a closed or open profile.
- `Profile_swept_shape`: This is a 2D line which is combined with a profile to create the shape of the profile feature. Data for this entity are depth and direction, which is always towards the z-axis.

To manufacture this feature in the turn-mill machine, the analyser needs to read the `feature_boundary` and according to the profile type chose the correct feature and operation. As stated above, the profile can be closed or open.

`Open_profile`: a type of profile that is an outline of shape with no enclosing or confining bounds. There are different types of open_profile: linear, `square_u`, `rounded_u tee`, `vee`, `partial_circular` and general profile.

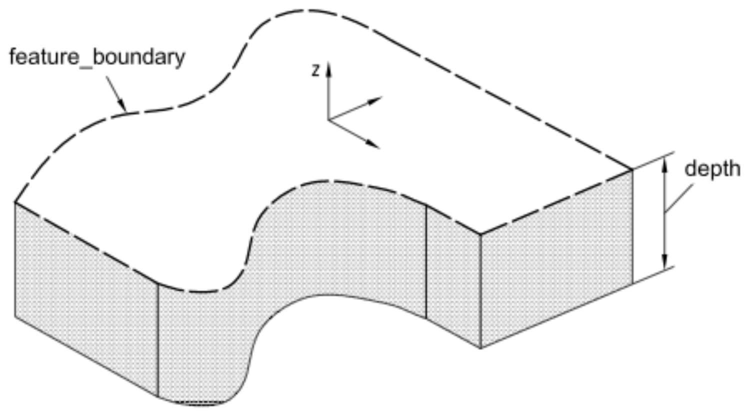


Figure 6.8: General outside profile adapted from (ISO 14649-10, 2002)

All types of `open_profile` will remain the same as features in source machine (milling machine) and for manufacturing these features milling operation will be used in the turn-mill machine.

```
If the milling feature = profile_feature
and the profile_feature = open_profile
then the turn-mill feature = profile_feature (same as milling)
and the turn-mill operation = milling operation
which is the milling_operation = bottom_and_side_milling
end if
```

`Closed_profile`: a type of profile that is an outline or shape that bounds an enclosed area with no opening.

There are four types of `closed_profile`: rectangular, circular, ngon and general closed profile. For each of these closed profiles the analyser needs to choose compatible features.

Rectangular closed profile:

```
If the closed_profile = rectangular_closed_profile
then the manufacturing_feature = milling feature
      the milling feature = general_outside_profile
      which all the data is same as source data
      the machining_operation in turn-mill machine =
milling_operation
      the milling_operation = bottom_and_side_milling
end if
```

Circular closed profile: if there is a circular profile located in the middle of the workpiece then the feature will be `outer_diameter` in turning. If it is not in the middle then the feature will be the same as the original one in the milling.

The operation for the `outer_diameter` feature is a turning operation and for a profile feature with a circular profile a milling approach is used in the turn-mill machine. Figure 6.9 shows the `outer_diameter` in turning technology.

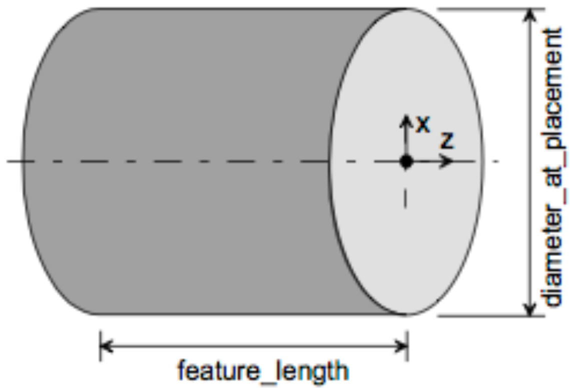


Figure 6.9: Outer diameter adapted from (ISO 14649-12, 2004)

The entity of the outer diameter is the same as the outer diameter with a taper, which is the taper, is cylindrical and to distinguish this in the feature entity, the `reduced_size` entity needs to be null.

```
If the closed_profile = circular_closed_profile
      if circular_closed_profile position = center
      then the manufacturing_feature = turning feature
      the turning feature = outer_diameter
      which feature_length = depth
      and diameter_at_placement = diameter of circular
      the machining_operation = turning
      the turning_machining_operation = contouring

      Else
      is it possible to position it to the center

      if yes
      then the manufacturing_feature = turning feature
      the turning feature = outer_diameter
      which feature_length = depth
      and diameter_at_placement = diameter of circular
      the machining_operation = turning
      the turning_machining_operation = contouring

      else
      the manufacturing_feature = milling feature
      the milling feature = general_outside_profile
      which all the data is same as source data
```

```
        the machining_operation = milling
        the milling_machining_operation = bottom_and_side_milling
        end if
        end if

End if
```

Ngon profile: is an enclosed area bounded by three or more connected straight line sides, as illustrated in Figure 6.10.

```
If the closed_profile = ngon_profile
then the manufacturing_feature = milling feature
      the milling feature = general_outside_profile
      which all the data is same as source data
      the machining_operation in turn-mill machine =
milling_operation
      the milling_operation = bottom_and_side_milling
end if
```

General profile: is an enclosed area bounded by an arbitrary shape.

```
If the closed_profile = general_closed_profile
then the manufacturing_feature = milling feature
      the milling feature = general_outside_profile
      which all the data is same as source data
      the machining_operation in turn-mill machine =
milling_operation
      the milling_operation = bottom_and_side_milling
end if
```

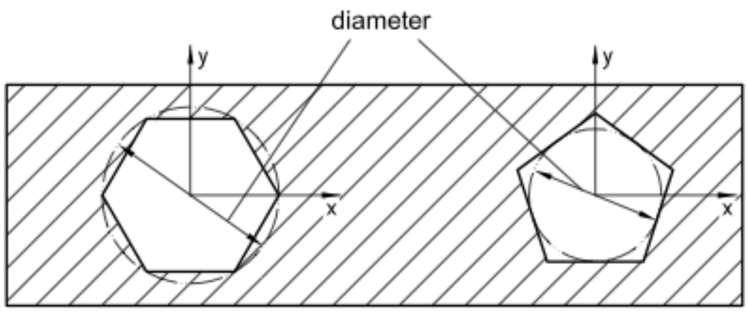


Figure 6.10: Ngon profile adapted from (ISO 14649-10, 2002)

(b) Shape profile

A shape profile is the removal volume of the shaped profile from the boundary shape of a workpiece. Floor condition limits the bottom of the boundary shape. The shape profile is an abstract supertype of `general_shape_profile`, `partial_circular_shape_profile`, `circular_closed_shape_profile` and `rectangular_open_shape_profile`. Figure 6.11 illustrates shape profile with its properties.

The information in the shape profile entity includes:

- `Floor_condition`: specification of the shape of the bottom,
- `Removal_direction`: direction of removal material.

To manufacture this feature in the turn-mill machine, the analyser needs to identify the type of `shape_profile`, if it is `general_shape_profile`, `partial_circular_shape_profile`, or `rectangular_open_shape_profile` then the operation remain same as milling, but if it is `circular_closed_shape_profile` then the operation will be turning.

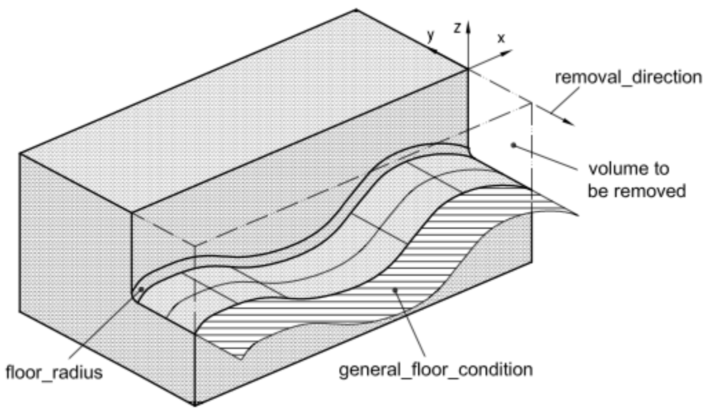


Figure 6.11: Shape profile adapted from (ISO 14649-10, 2002)

```
If the shape_profile = circular_closed_shape_profile
      then check the position
      if circular_closed_shape_profile position = center
      then the manufacturing_feature = turning feature
      the turning feature = outer_diameter
      which feature_length = depth
      and diameter_at_placement = diameter of circular
      the machining_operation = turning
      the turning_machining_operation = contouring

      Else
      is it possible to position it to the center

      if yes
      then the manufacturing_feature = turning feature
      the turning feature = outer_diameter
      which feature_length = depth
      and diameter_at_placement = diameter of circular
      the machining_operation = turning
      the turning_machining_operation = contouring

      else
      the manufacturing_feature = milling feature
      the milling feature = general_outside_profile
      which all the data is same as source data
      the machining_operation = milling
      the milling_machining_operation = bottom_and_side_milling
      end if


else
the manufacturing_feature = milling_feature
the machining_operation = milling
end if

End if
```

6.3.3 Round hole

Round hole is a feature that is used in most CNC technologies. The hole is identified by its centre point at the surface and located at x = y = 0. In this feature the bottom of the hole is not considered in the hole's depth. Figure 6.12 shows the different types of holes.

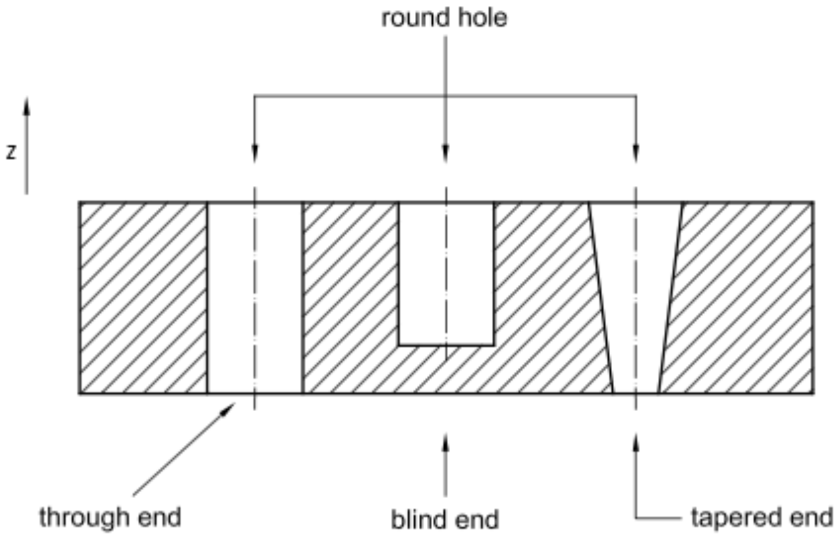


Figure 6.12: Hole types adapted form (ISO 14649-10, 2002)

The information in the `round_hole` entity is:

- `Diameter`: the diameter of the hole,
- `Change_in_diameter`: this parameter is used to specify a hole with a taper,
- `Bottom_condition`: specifies the bottom of the hole.

This feature will be same in both milling and turn-mill technology but in the latter the operation used is drilling.

```
If the machining_feature = round_hole
then feature for milling and turn-mill machine = round_hole
      and the machining_operation in turn-mill machine =
milling_operation
which is the milling_operation = drilling_operation

end if

if the centre of the hole is located on (x = z = 0) or (y = z = 0)
then the depth will be negative y or x.
end if
```

6.3.4 Slot

Typically, a slot is manufactured by a single sweep of a tool along the core of travel. Slot has a number of different entities such as: `course_of_travel`, `swept_shape` and `end_conditions`. Figure 6.13 illustrates two slots, one open end type and one radiused end type.

The information regarding the `slot` entity is:

- `Depth`: this identifies the depth of the slot from the `machining_feature` entity.
- `Course_of_travel`: this entity identifies the path that a tool needs to travel to manufacture the slot. The path also identifies the centre of the slot.
- `Swept_shape`: the cross-section generated by the tool.
- `End_conditions`: this entity describes the end condition of the slot which may be an open end, or a closed end, such as woodruff, radiused, flat or looped.

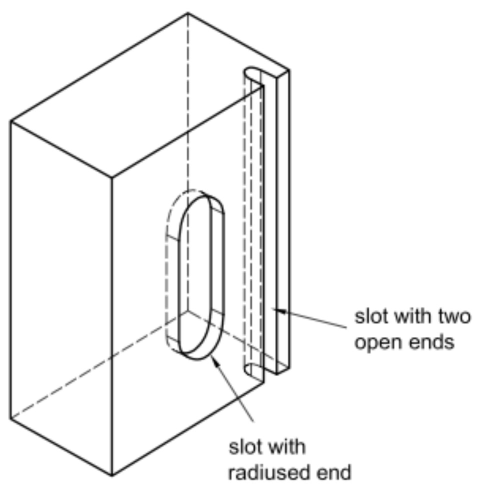


Figure 6.13: Slots adapted from (ISO 14649-10, 2002)

To manufacture such a feature in a turn-mill machine, the `end_conditions` of the slot need to be checked so that the correct feature is generated in the turn-mill machine. The analyser will check the end condition of the slot and if the end condition is a loop end condition then the feature is groove, which is one of the revolving features in turning features. If the shape of the groove is identical to the shape of the tool then the feature will be `cut_in` feature, which is one of the turning specific features. For the rest of the `end_conditions` in slot, the feature will be the same as the one in the milling technology and the operation will be milling.

Groove: is one type of revolved feature with the elements identifying the feature being: radius, material side and sweep. Figure 6.14 shows a groove in two different directions.

For groove, the information in the slot entity is:

- `Radius`: the distance from the axis of rotation to define the placement of the profile that will be swept about the axis.
- `Material_side`: the direction of material that will be cut by the tool.
- `Sweep`: an outline or shape that shall be revolved about an axis.

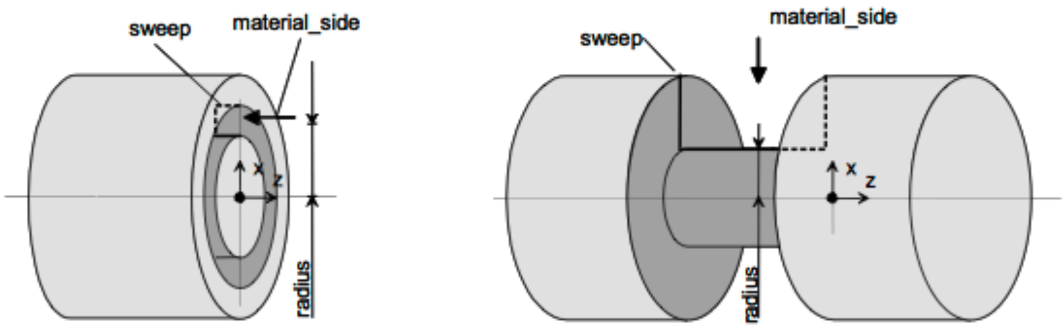


Figure 6.14: Groove adapted from (ISO 14649-12, 2004)

Cut in: is a kind of groove or slot where the geometrical shape of the groove is identical to the shape of the used tool. This feature has two entities: `depth` and `cut_in_direction`. Figure 6.15 shows the cut in feature.

```
If the machining_feature = slot
then check the end_conditions
      if the end_conditions = loop_slot_end_type
      then check the direction of course_of_travel
            if the direction = (0,0,1)
                  then check the position
                  if the position = centre of workpice
                  then the machining_feature = turning feature
                  turning feature = groove
                  depth of the groove = depth of slot
                  sweep = swept_shape
                  radius of the groove = radius of the travel_path
                  material_side = (0,0,1)
                  else machining_feature = slot
                  end if
            else (the direction ≠(0,0,1))
            then the machining_feature = turning feature
            turning feature = groove
            radius of the groove = depth of slot
            sweep = swept_shape
            material_side = (1,0,0)
            end if
      else machining_feature = slot
      end if
else the machining_feature = slot
end if

If the groove shape = tool shape
then the turning feature = cut_in
which the  depth = depth of slot
and cut_in_direction = material_side
end if

if the turning_feature = groove
then the turning operation = grooving
else machining operation = milling
which milling_operation = bottom_and_side_milling
end if

if the turning_feature = cut_in
then the turning operation = ctting_in
end if
```

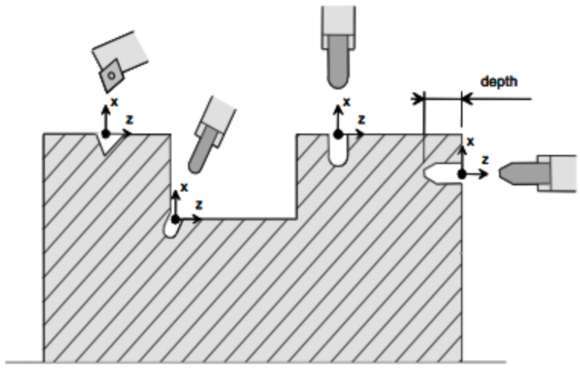


Figure 6.15: Cut in adapted from (ISO 14649-12, 2004)

6.3.5 Step

Step is a `machining_feature` that is a volume of removal material from the bottom and side of the workpiece. Figure 6.16 illustrates the step. Step attributes are: `open_boundary`, `wall_boundary` and `its_boss`.

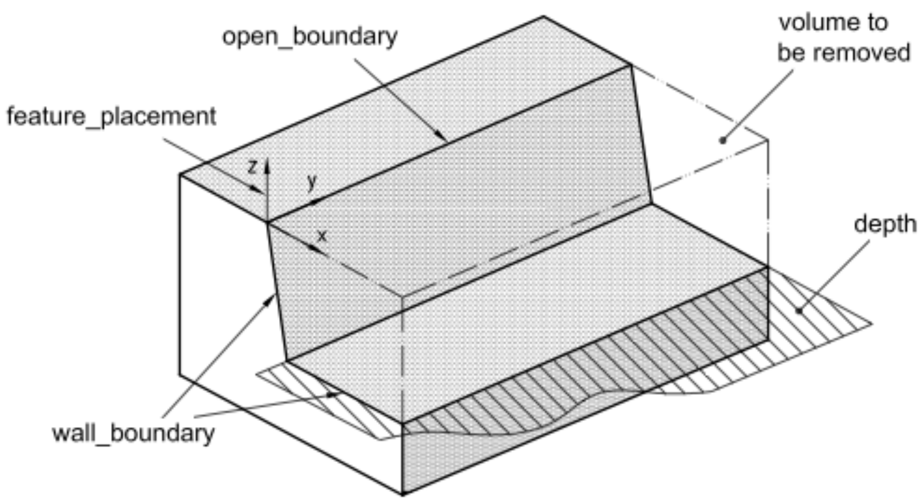


Figure 6.16: Step adapted from (ISO 14649-10, 2002)

The information in the `step` entity is:

- `Depth`: identifying the depth of the slot from the `machining_feature` entity,
- `Open_boundary`: an outline or shape that forms the upper edge of the step where the cut material will be on the right of the curve,
- `Wall_boundary`: identifying whether the wall boundary is V shaped or not,
- `Its_boss`: identifying whether or not there is any boss in the step.

For manufacturing step in both technologies the feature will remain same and the operation is milling for both technology.

```
If the milling feature = step
then the turn-mill feature = step
and the turn-mill operation = milling operation
which is the milling_operation = bottom_and_side_milling
end if
```

6.3.6 Pocket

Pocket is a machining feature and can either be an open or closed pocket. The geometry of the pocket is identified by its contour on the outer face and its depth. The pocket entities are: `its_boss`, `slope`, `bottom_condition`, `planar_radius` and `orthogonal_radius`. Figure 6.17 shows the open and closed pockets.

The information in the pocket entity is:

- `Depth`: identifying the depth of the slot from the `machining_feature` entity,
- `Its_boss`: if there is any boss in the pocket then this part will not cut automatically during the manufacturing process.
- `Slope`: identifies the angle of the border of the pocket against the z-axis (it is zero by default).

- `Bottom_condition`: identifies the bottom condition of the pocket which can be `through_pocket_bottom_condition`, `planar_pocket_bottom_condition`, `radiused_pocket_bottom_condition` and `general_pocket_bottom_condition`.
- `Planar_radius`: the planar radius of a fillet.
- `Orthogonal_radius`: the orthogonal radius of a fillet.

If the pocket is a closed pocket then:

- `Feature_boundary`: a shape that describes the upper edge of the pocket, which can be `rectangular_closed_profile`, `circular_closed_profile`, `ngon_profile` and `general_closed_profile`.

And if the pocket is open pocket then:

- `Open_boundary`: a shape that describes the upper edge of the open pocket.
- `Wall_boundary`: a shape that describes the side edge of the open pocket.

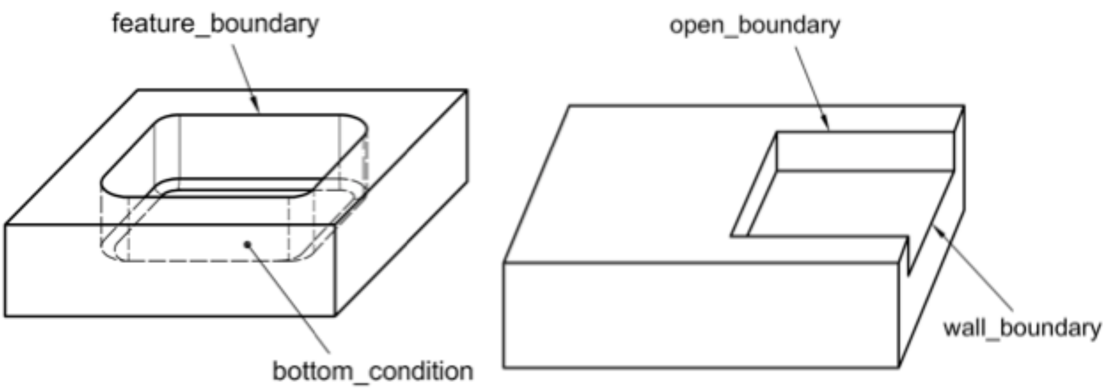


Figure 6.17: Closed pocket (left) and open pocket (right)
adapted from (ISO 14649-10, 2002)

To manufacture this feature, if there is a feature boundary in a closed pocket then the analyser needs to choose the correct feature in the turn-mill machine. If the pocket is an open pocket then the feature is the same and the operation is milling, but if the pocket is closed and the feature boundary is circular without any boss then the feature can be turning according to the position of the feature.

```
If the machining_feature = pocket
then check the pocket type
      if the pocket = open_pocket
      then the feature for both technology = open_pocket
      and operation in turn-mill technology = milling
      else (which is the pocket = closed_pocket)
      then check the feature_boundary
            if the feature_boundary = circular_closed_profile
            and if its position = center
            and its_boss = 0
            then the machining_feature = turning feature
            turning feature = groove
            depth of the groove = depth of pocket
            sweep = square_u_profile
                  which first_radius = second_radius = planar_radius
                  and first_angle = second_angle = 0
                  and width = diameter of the circular / 2
            radius of the groove = 0
            material_side = (0,0,1)
            else machining_feature = slot
            end if
 end if
```

6.4 The user interface of the XTSys prototype

The prototype shown in Figure 6.18 accepts STEP-NC codes in a text file as the input for the system.

The selected file will be opened in the milling STEP-NC code textpane. The imported code screenshot is illustrated in Figure 6.19.

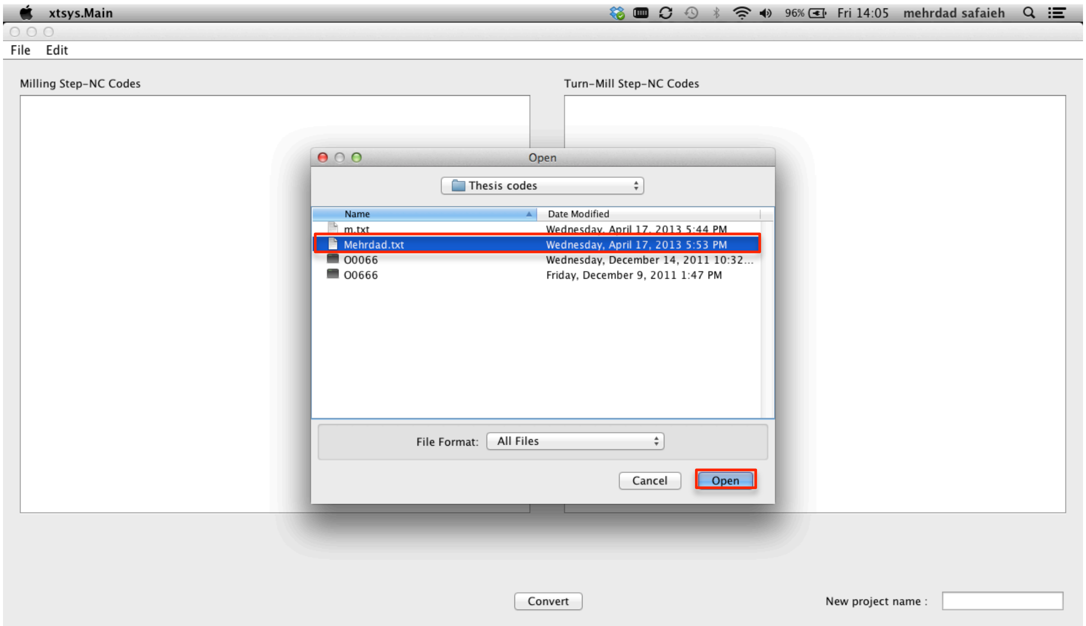


Figure 6.18: Overview of XTSys screen shot

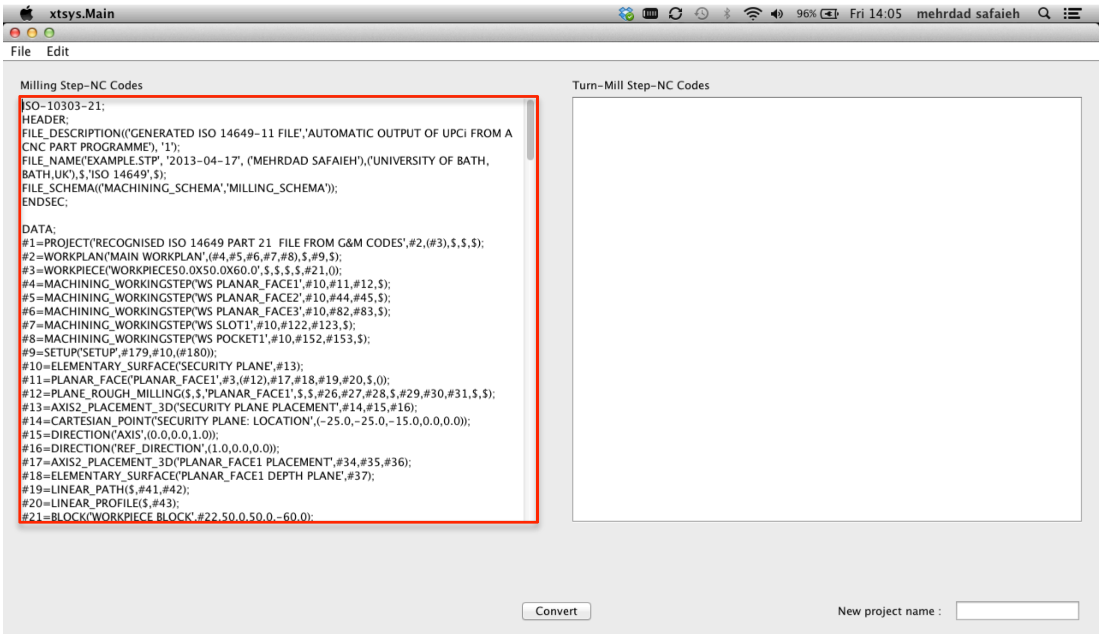


Figure 6.19: Imported milling STEP-NC codes to XTSys

To generate new STEP-NC code for the turn-mill machine, first a new name for the project needed and then by selecting convert new code it will be generated in the turn-mill STEP-NC textpane. Figure 6.20 shows the screenshot of this scenario.

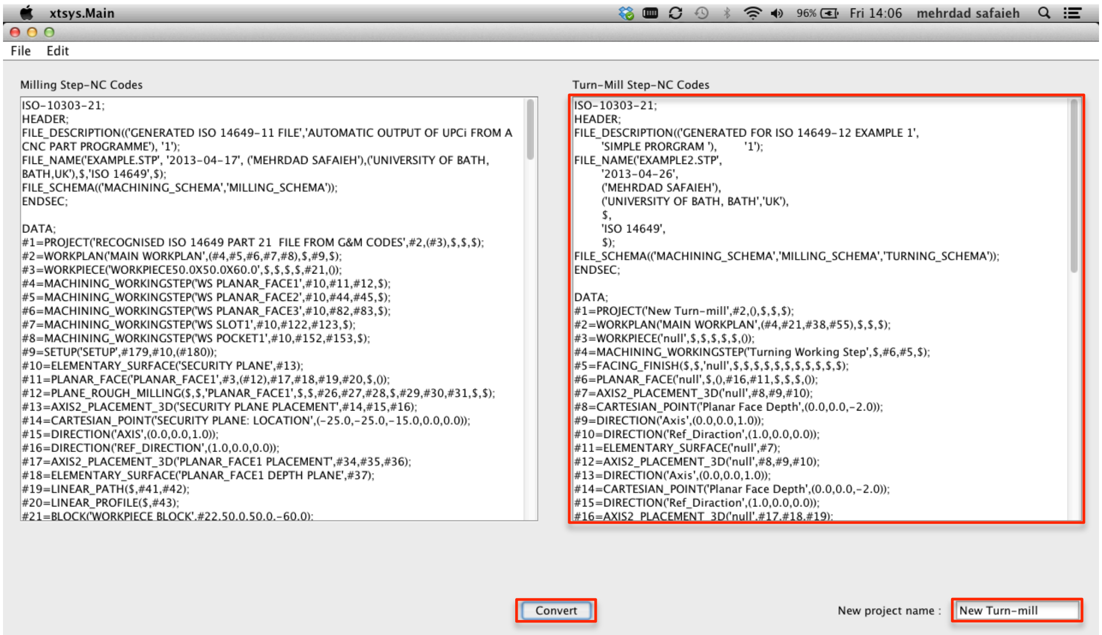


Figure 6.20: Generated turn-mill STEP-NC code

7 Experimental verification of the XTSys

7.1 Introduction

In this chapter, in order to verify the XTSys system three example parts have been utilised to illustrate interoperability for CNC machines with different technologies within the limitations of the prototype.

The example parts have been designed based on a number of features that can be machined with CNC machines with milling and turn-mill technologies and represent the three classes of parts defined in 3.6.

The example parts were programed for a milling machine and converted to turn-mill code by XTSys. The programmes were written for a Dugard Eagle 850 4-axis milling centre with a Fanuc series 18i-MB controller. The code was then converted by XTSys to allow the same parts to be machined by a Hyundai-Kia SKT 15LM turn-mill centre with a Fanuc series 21i-TB controller.

7.2 Experimental results

Test part one:

Test part 1 illustrated in figure 7.1 was chosen in this research to show the basic functionality of the framework prototype. The part was cut from a round bar with a diameter of 50 mm and a height of 60 mm and was designed with five different features which are machineable by both machines. Part drawing is provided in appendix C.

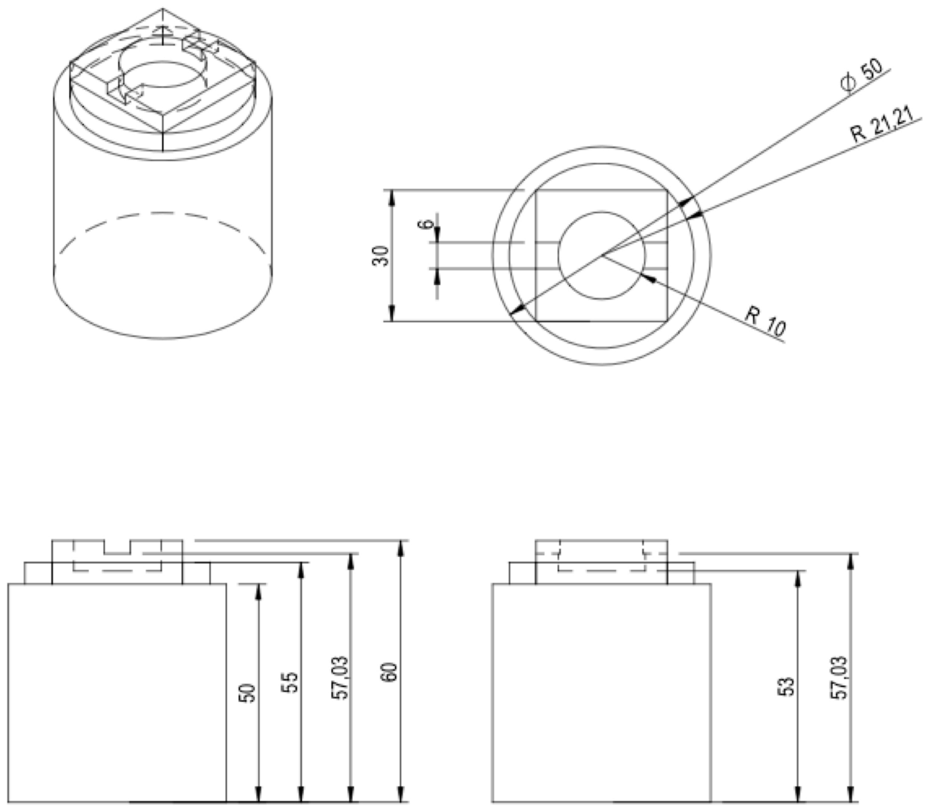


Figure 7.1: Test part 1 drawing

The test part 1 encompasses a number of manufacturing features and operations, some of which are the same on both machines and some of which are different. The features and operations that made up the process plan in the source machine (the milling centre) are as follows:

i. `Planar_Face`: The machining of the test part begins with a facing operation that is `Plane_rough_milling` and `Plane_finish_milling` operation in STEP-NC, as shown in figure 7.2.

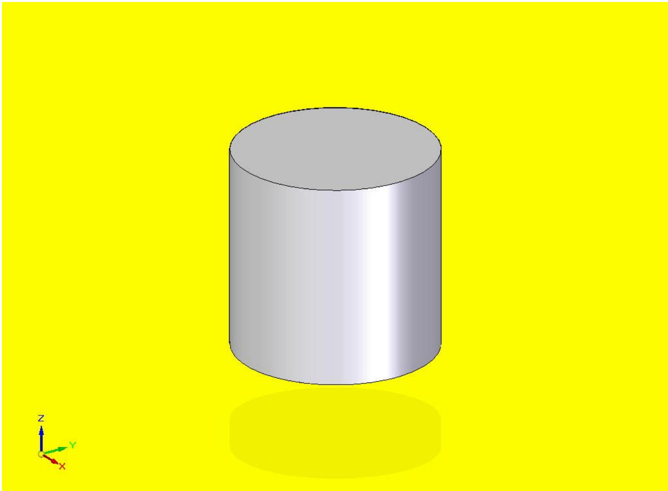


Figure 7.2: Planar_Face

ii. `Profile_Feature`: To create the `profile_feature` in the test component, bottom and side milling operation are used, as shown in figure 7.3.

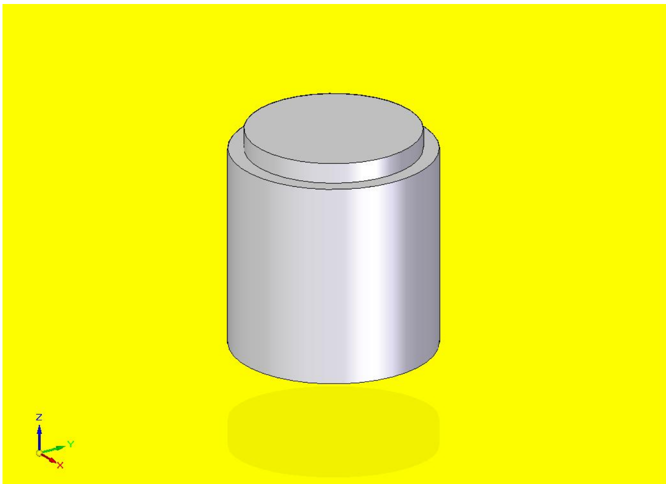


Figure 7.3: Profile_Feature

iii. `Profile_Feature`: For machining `Profile_Feature` feature in the test part, once again `bottom_and_side_milling` operations are used, as shown in figure 7.4.

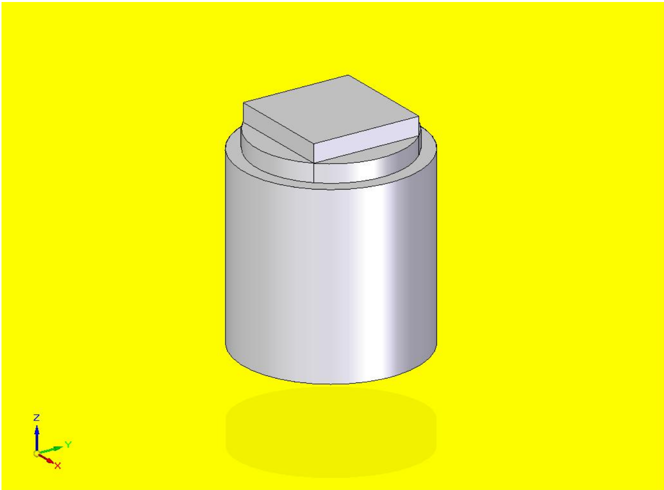


Figure 7.4: Profile_Feature

iv. `Closed_Pocket`: To create a circular pocket in the test component, `bottom_and_side_milling` is used, as shown in figure 7.5.

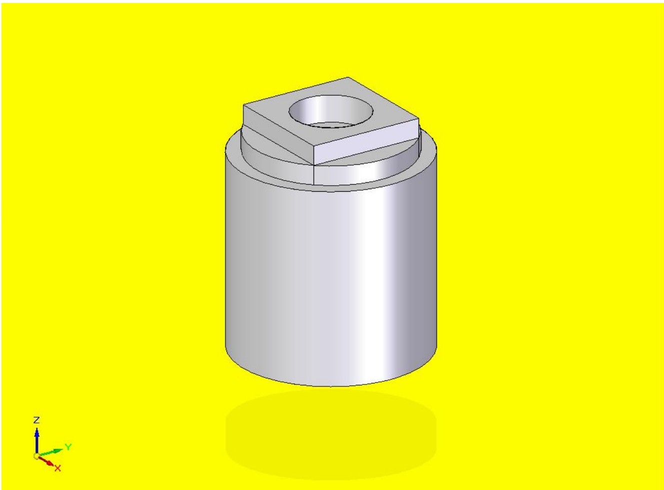


Figure 7.5: Circular pocket

v. Slot: to create a slot in the test part bottom_and_side_milling is used, as shown in figure 7.6.

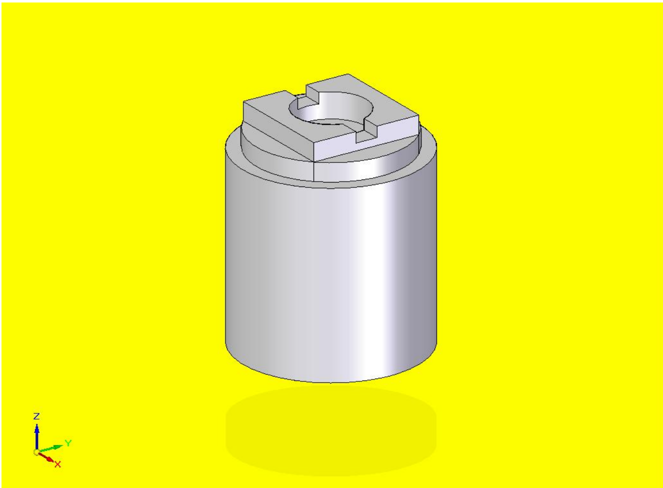


Figure 7.6: Slot

To machine the test part 1 in the Dugard Eagle 850 three tools were used, a 66 mm face mill, a 12 mm slot drill and a 6 mm slot drill. Figure 7.7 sets out which tool was used for machining each feature, together with the specific operations utilised. The resulting machined part is shown in figures 7.8 and 7.9.

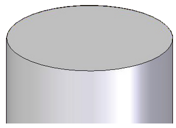
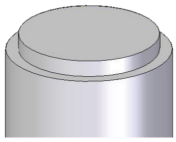
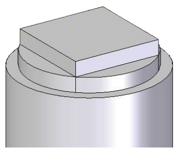
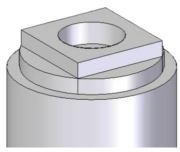
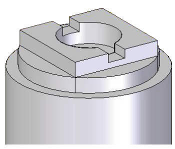
Tools		Features	Operations	
	66mm face mill	Planar_Face		Plane_rough_milling & Plane_finish_milling
	12mm slot drill	Profile_Feature		Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
	12mm slot drill	Profile_Feature		Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
	6mm slot drill	Closed_Pocket		Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
	6mm slot drill	Slot		Bottom_and_side _rough_milling & Bottom_and_side _finish_milling

Figure 7.7: Tools, features and operations in milling technology for part 1

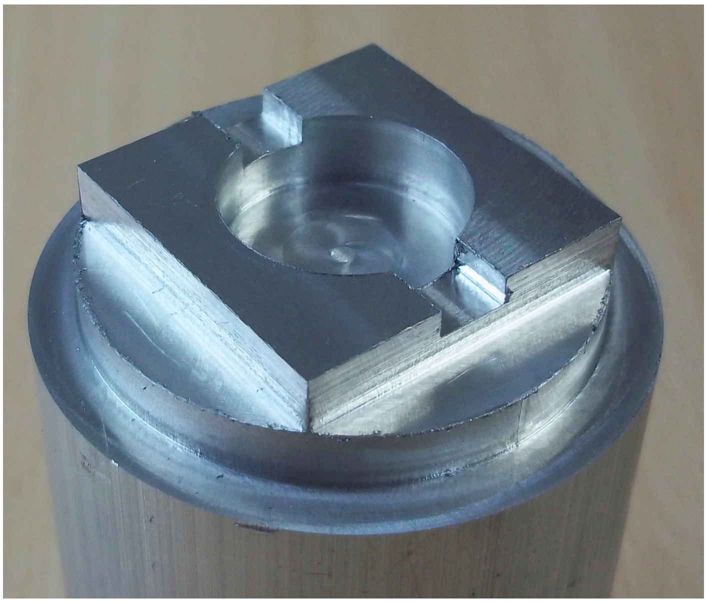


Figure 7.8: Milling finished part

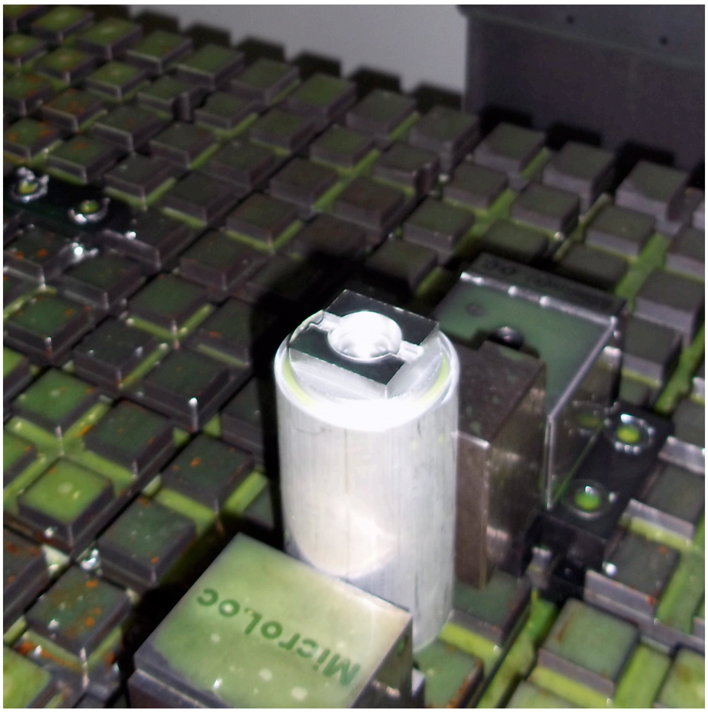


Figure 7.9: Milling part in the machine after machining

To machine the experimental part with the Dugard milling machining centre, the G&M codes can be generated manually or automatically by using CAD/CAM software. In the research, the G&M codes were generated manually. The generated code was transferred to a STEP-NC file using an UPCi converter, which read the G&M codes and converted them to a STEP-NC format. The G&M codes are shown in appendix B 12.1 and 12.2.

The converted STEP-NC file was input into the XTSys system in order to generate a new STEP-NC file for turn-mill technology. This would then be used for machining the test part with a Hyundai-Kia SKT 15 LM turn-mill centre. The XTSys system read the source STEP-NC file and, by using information from the destination machining centre, generated the new STEP-NC file. Figure 7.10 shows these conversions. The completed listing of the source STEP-NC can be seen in appendix B 12.3 and the destination STEP-NC in appendix B 12.4.

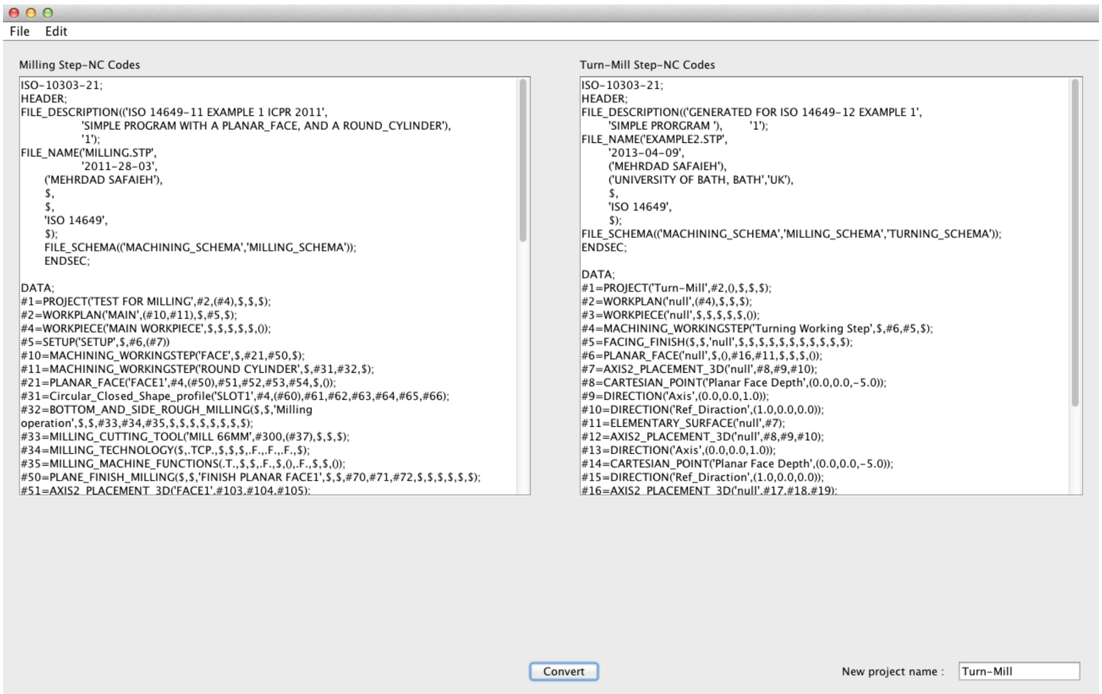


Figure 7.10: Conversion in XTSys

In the experimental part, the new code for the turn-mill machining centre had different operations for `Planar_Face` and `Profile_Feature`. In the original milling technology process, the operation for `Planar_Face` was plane milling whereas, when the task was translated to turn-mill technology, XTSys chose a facing operation taking into account the availability of technology and tools in the destination machine.

Likewise a contouring operation was chosen for machining `Profile_Feature` in the destination machine. The rest of the features used in the destination turning machine were the same as those used in the original milling operations: that is, bottom and side milling was used in both machines. Figure 7.11 shows the conversion of operations and tools in the XTSys analyser.

The result of using these tools and operations is shown in figures 7.13 and 7.14.

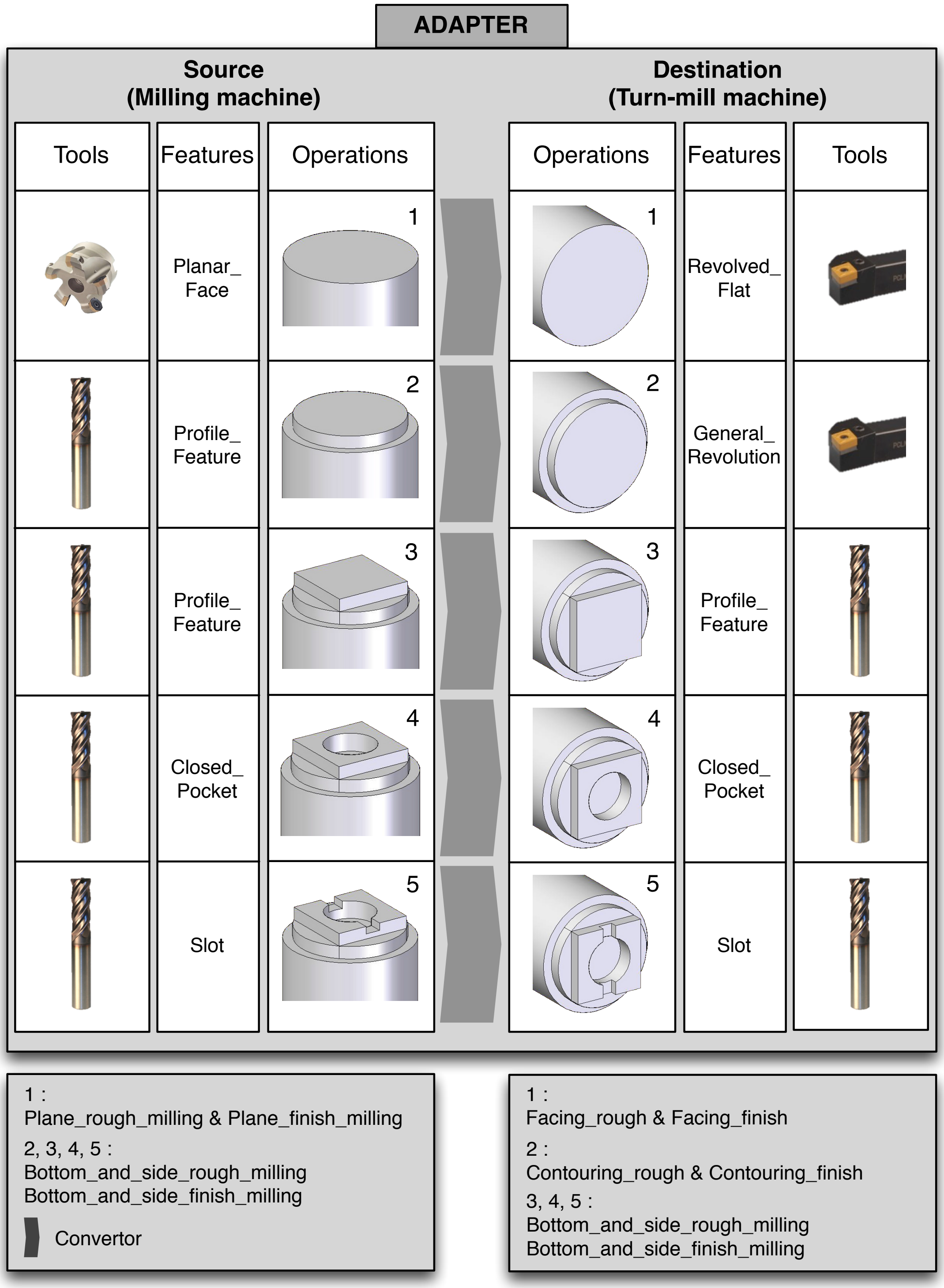


Figure 7.11: Features, operations and tools for the test part 1 in the XTSys adapter

The output information for the turn-mill machining centre is illustrated in figure 7.12.

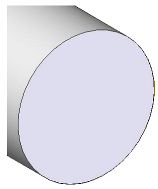
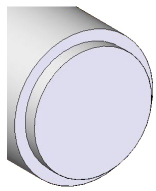
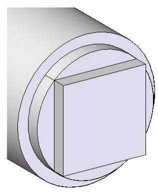
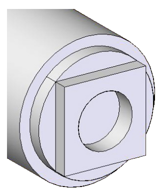
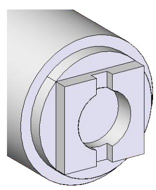
Tools	Features	Operations
 profile rough	Revolved_Flat	 Facing_rough & Facing_finish
 profile rough	General_Revolution	 Contouring_rough & Contouring_finish
 6mm slot drill	Profile_Feature	 Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
 6mm slot drill	Closed_Pocket	 Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
 6mm slot drill	Slot	 Bottom_and_side _rough_milling & Bottom_and_side _finish_milling

Figure 7.12: Tools, features and operations in turn-mill technology for part 1

Figure 7.13: Finished turn-mill part

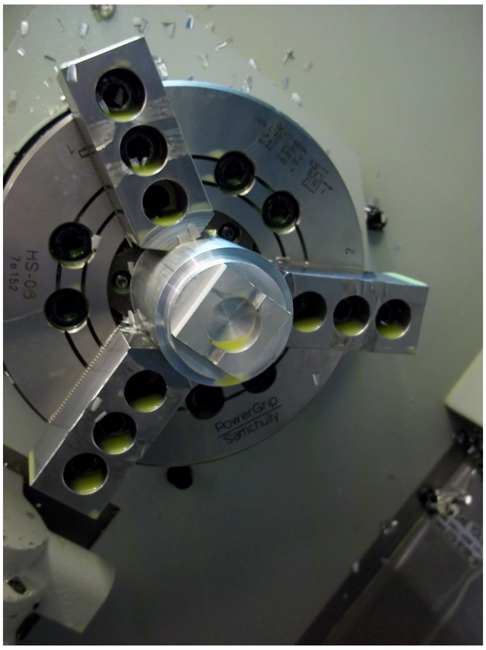


Figure 7.14: Experimental part in the turn-mill machining centre for part 1

Test part two:

Test part two represents a more complicated set of features. The source and destination STEP-NC file can be seen in appendix B 12.5 and 12.6. Following the same scenario for part two (illustrated in Figure 7.15), the new code for the turn-mill machining centre had different operations and features, `Planar_Face` in milling is `Revolved_Flat` and one of the `Profile_Feature` is `Geberal_Revolution`. In the original milling technology process, the operation for `Planar_Face` was plane milling whereas, when the task was translated to turn-mill technology, XTSys chose a facing operation taking into account the availability of technology and tools in the destination machine. Likewise a contouring operation was chosen for machining `Profile_Feature` in the destination machine. The rest of the features used in the destination turning machine were the same as those used in the original milling operations: that is, bottom and side milling was used in both machines.

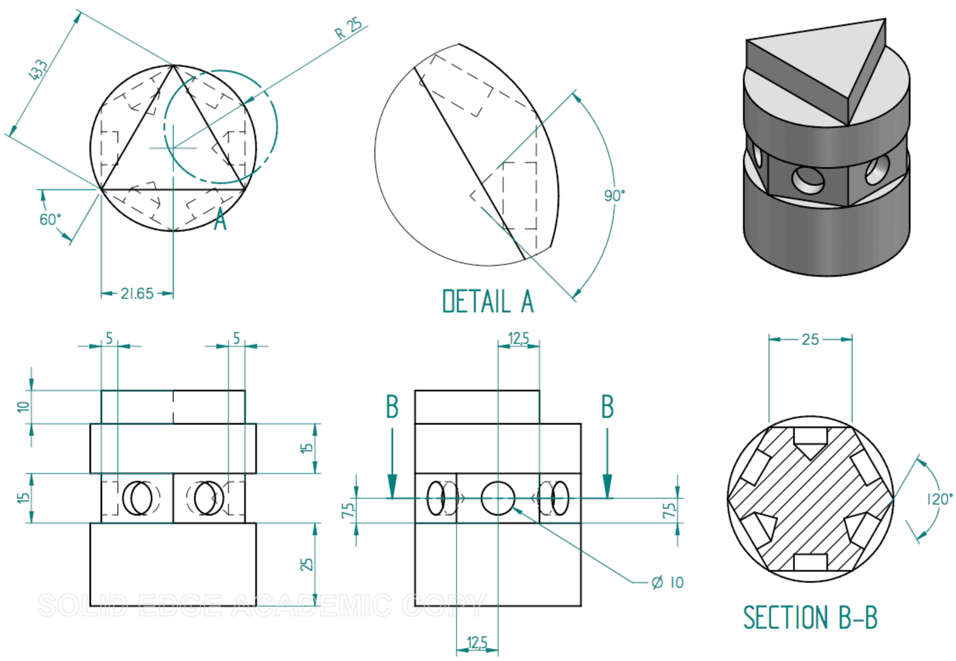


Figure 7.15: Test part 2 drawing

Figure 7.16 shows the conversion of operations and tools in the XTSys analyser. For this specific part, if 3-axis milling machine is chosen then for machining the part, multi setups are needed: one for features on z direction and six for features on x direction. If a 4-axis machining centre with a rotary table is chosen then the machining can be done in one setup. In this research, the generated STEP-NC code is based on one setup for the 4-axis milling machine.

Figure 7.17 (milling machine) and Figure 7.18 (turn-mill machine) illustrate the tools, features and operations in each technology.

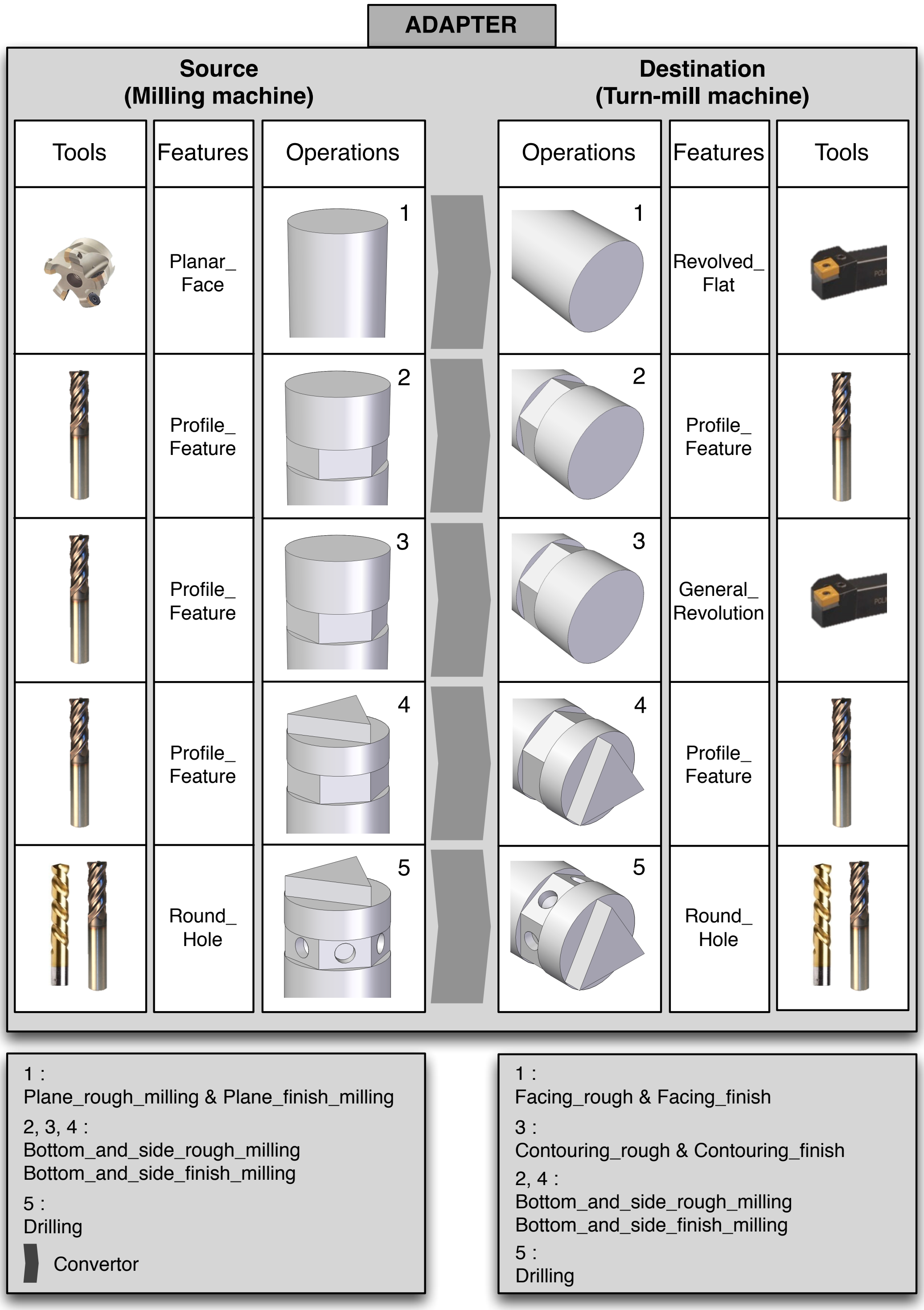


Figure 7.16: Features, operations and tools for the test part 2 in the XTSys adapter

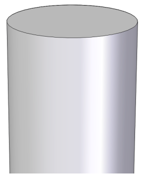
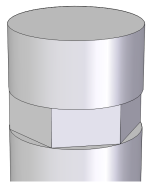
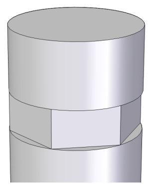
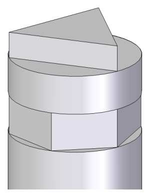
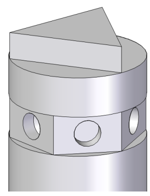
Tools	Features	Operations
 66mm face mill	Planar_Face	 Plane_rough_milling & Plane_finish_milling
 12mm slot drill	Profile_Feature	 Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
 12mm slot drill	Profile_Feature	 Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
 6mm slot drill	Profile_Feature	 Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
 10mm slot drill & 10mm twist drill	Round_Hole	 Drilling

Figure 7.17: Tools, features and operations in milling technology for part 2

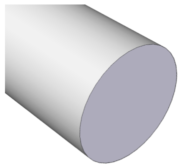
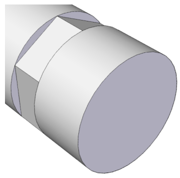
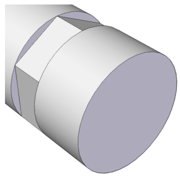
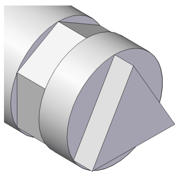
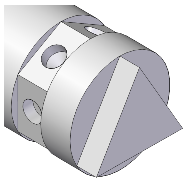
Tools	Features	Operations
 profile rough	Revolved_Flat	 Facing_rough & Facing_finish
 12mm slot drill	Profile_Feature	 Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
 profile rough	General_Revolution	 Contouring_rough & Contouring_finish
 6mm slot drill	Profile_Feature	 Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
 10mm slot drill & 10mm twist drill	Round_Hole	 Drilling

Figure 7.18: Tools, features and operations in turn-mill technology for part 2

Test part three:

In part three (which is shown in Figure 7.19), the new code for the turn-mill machine had different operations and features, `Planar_Face` in milling is `Revolved_Flat` and couple of the `Profile_Features` are `Geberal_Revolution`. The source and destination STEP-NC file can be seen in appendix B 12.7 and 12.8. This part has more complex features than the other two and because of this the number of setups will be different according to the number of axis in the machining centre. In the original milling technology process, the operation for `Planar_Face` was plane milling whereas, when the task was translated to turn-mill technology, XTSys chose a facing operation taking into account the availability of technology and tools in the destination machine. Likewise a contouring operation was chosen for machining `Profile_Feature` in the destination machine.

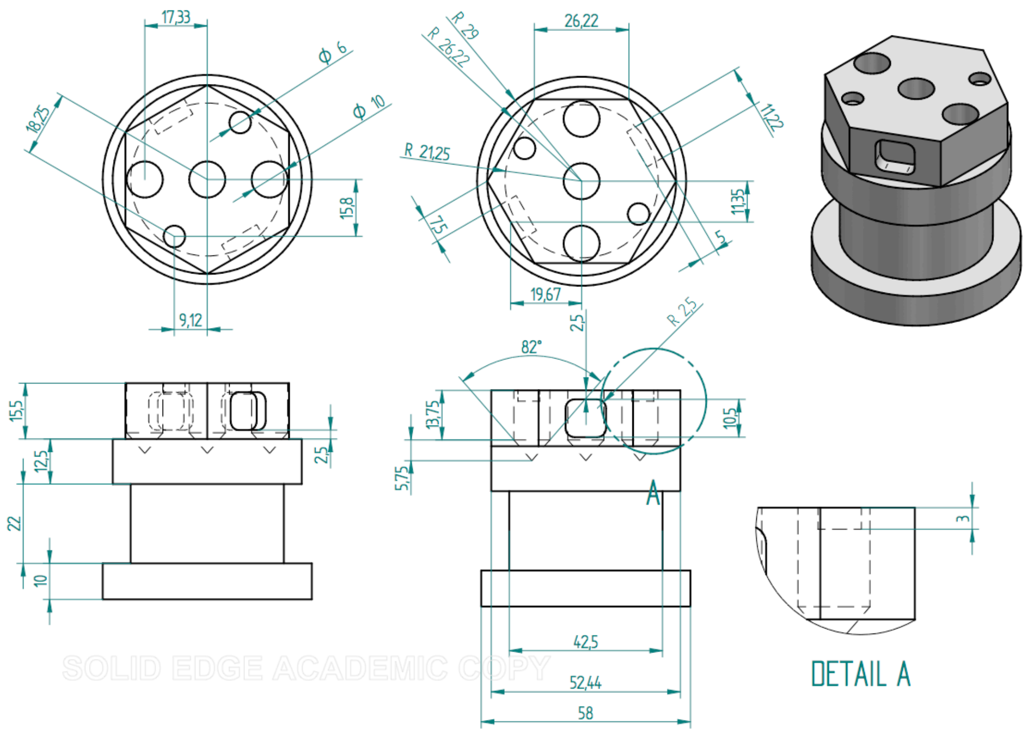


Figure 7.19: Test part 3 drawing

The rest of the features used in the destination turning machine were the same as those used in the original milling operations: that is, `bottom_and_side_milling` was used in both machines. Figure 7.20 shows the conversion of operations and tools in the XTSys analyser. For this specific part, if 4-axis milling machine is chosen then for machining the part need to be done in 2 setups, one for features on z direction and one for features on x direction.

If 3-axis machining centre with rotary table is chosen then the machining setup still need to be done by two setup because of the round holes in the part. For machining this part in one setup then a 5-axis machine can be used. In this research, the generated STEP-NC code for this part is based on 4-axis machining centre which capable of machine all the features in two setups.

Figure 7.21 (milling machine) and Figure 7.22 (turn-mill machine) illustrates the tools, features and operations in each technology.

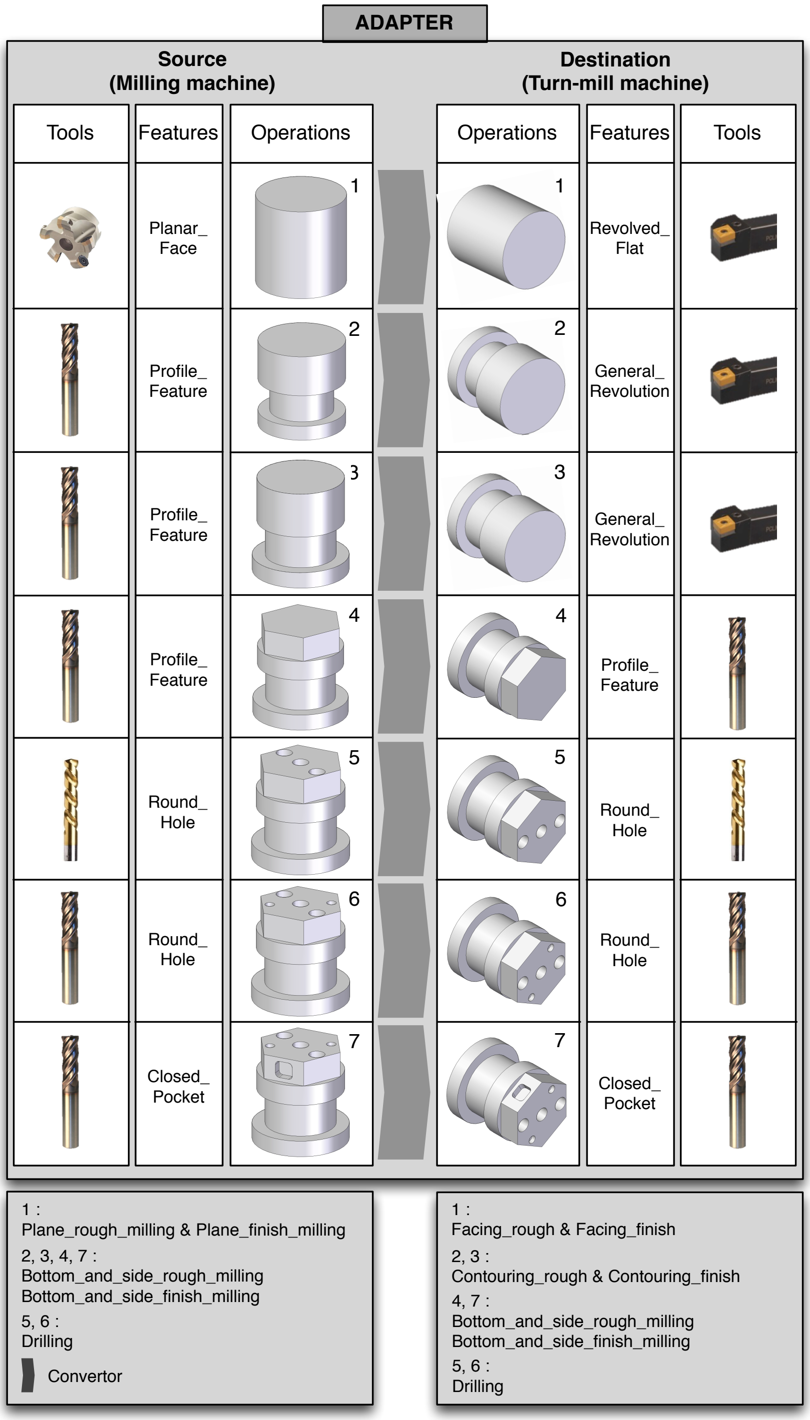


Figure 7.20: Features, operations and tools for the test part 3 in the XTSys adapter

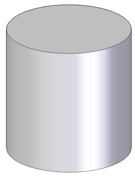
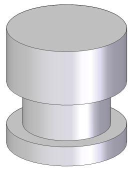
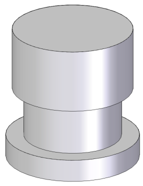
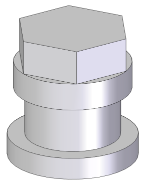
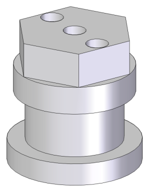
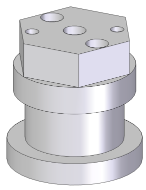
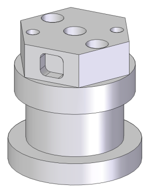
Tools		Features	Operations	
	66mm face mill	Planar_Face		Plane_rough_milling & Plane_finish_milling
	12mm slot drill	Profile_Feature		Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
	12mm slot drill	Profile_Feature		Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
	10mm slot drill	Profile_Feature		Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
	10mm drill	Round_Hole		Drilling
	6mm slot drill	Round_Hole		Drilling
	6mm slot drill	Closed_Pocket		Bottom_and_side _rough_milling & Bottom_and_side _finish_milling

Figure 7.21: Tools, features and operations in milling technology for part 3

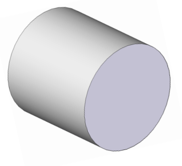
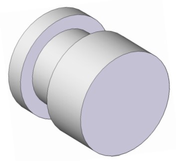
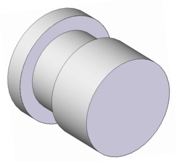
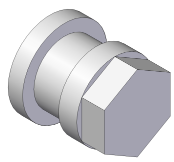
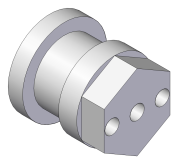
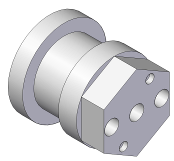
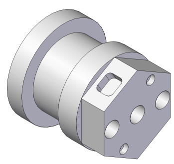
Tools	Features	Operations
 profile rough	Revolved_Flat	 Facing_rough & Facing_finish
 profile rough	General_Revolution	 Contouring_rough & Contouring_finish
 profile rough	General_Revolution	 Contouring_rough & Contouring_finish
 10mm slot drill	Profile_Feature	 Bottom_and_side _rough_milling & Bottom_and_side _finish_milling
 10mm drill	Round_Hole	 Drilling
 6mm slot drill	Round_Hole	 Drilling
 6mm slot drill	Closed_Pocket	 Bottom_and_side _rough_milling & Bottom_and_side _finish_milling

Figure 7.22: Tools, features and operations in turn-mill technology for part 3

8 Discussion

8.1 Introduction

In this chapter a number of issues are discussed in relation to the scope of the research and the methods used in within the context of cross-technology interoperability.

8.2 Literature on CNC manufacturing and interoperability

The review of the literature on manufacturing of asymmetric cylindrical rotational parts with prismatic components, identified in chapter 3, has shown that asymmetric parts with prismatic components can be machined with different CNC machining centres with different technologies by using different tools and different CNC machine part programing codes. The review of interoperability for CNC manufacturing conducted in chapter 4, showed that transferring information between CAD/CAM/CNC systems is necessary and that there has been extensive research into ways to implement interoperability in CNC manufacturing. These two chapters also show the need for further research into interoperability between CNC machines with different technologies. It was also shown that transferring machining information without reprocessing by manufacturing experts between CNC machines with different technologies is currently not possible.

The development of STEP-NC standards for transferring information between CAD/CAM/CNC systems is a goal that is now closer to realisation but still there is a gap in transferring information between CNC machines.

It is argued in this study, therefore, that while extensive research has been conducted into interoperability in CAD/CAM systems, there is a gap in the research in respect to interoperability between CNC machines using the same standards that are used in CAD/CAM systems. Filling this gap offers the opportunity to push interoperability in manufacturing forward. Furthermore, by having a system that enables interoperability between CNC machines without the manual involvement of manufacturing experts will not only solve the interoperability issues but also provide the necessary information and methods to solve the future problem of fully automated manufacturing systems.

8.3 A novel approach for implementation of interoperability

The approach proposed in chapter 5 provides a novel foundation for implementing interoperability between two CNC machines with different technologies. The philosophy behind the framework is manufacturing the same part with different CNC machines using different technologies and without any need for input from manufacturing experts. This is different from the existing way of machining a part that has features which are capable of being manufactured with different CNC technologies, where there is a need for a manufacturing expert to reprocess the machine code for the new machine. In the new approach that is proposed in chapter 5, an adapter will read the code from one technology and then generate the new code for the new technology automatically by using the information that is available for that new technology.

During the development of this novel method for implementation of interoperability, following issues were identified:

- The input of the XTSys is STEP-NC file, for machine parts with G&M codes there is a need for converting G&M codes to STEP-NC format. Converting G&M codes to STEP-NC format is still new and there are many problems involved in the process. These include tool selection, feature selection, operation selection and thus converting complex G&M codes for complex parts is still a challenge.

- The output of the XTSys system is a STEP-NC file which needs to be machined with CNC machines with controllers that support the STEP-NC format. At present, STEP-NC enabled intelligent CNC have not been commercially introduced. Therefore there is a requirement for translating STEP-NC to G&M codes.

- ISO 14649 provides individual data models for turning and milling. The combination of these technologies for turn-mill machines presents a challenge in combining data models for two parts of the standard together (ISO 14649-11 and ISO 14649-12).

8.4 Development of a prototype implementation of XTSys

The XTSys prototype has used STEP-NC information models within JAVA programming to transfer STEP-NC machine code from one technology to another. Furthermore, because of the object oriented approach in XTSys, programmers can add and edit the prototype in the future with minimal effort to include additional functionality.

XTSys uses a manufacturing dictionary, as described in 5.2.1, for reading, analysing and writing machine codes from all standards and this adaptability in relation to standards is a key advantage of XTSys. In the future, therefore, the XTSys system will be able to work with an even more varied range of technologies once more resources are added to its manufacturing dictionary.

The input and output of XTSys is a STEP-NC file, while in reality most machines work with G&M codes. The XTSys system therefore communicates with UPCi and iNet systems such that UPCi converts G&M codes to the STEP-NC format and then, at the end of the process, iNet converts STEP-NC back to a G&M code format. Through this communication it will be possible to machine old legacy parts with these new technological systems.

8.5 Experimental verification of XTSys

To evaluate the prototype implementation of XTSys, three asymmetric cylindrical rotational test parts were designed to be manufactured with a milling machining centre with the machine code then being translated to a new code that would work with a turn-mill machining centre. In this verification, first a G&M code of a test part is generated and then by using UPCi, the STEP-NC file for the milling machine is generated. There are a number of difficulties in generating the STEP-NC file for the test parts due to in compatibility of UPCi with some of the features used in test parts, the generated programmes, thus required some manual editing.

For converting the STEP-NC turn-mill output file to G&M codes for machining with the CNC Hyundai-Kia turn-mill machine, there were difficulties as there is a lack of software information for converting turn-mill technology form STEP-NC to G&M code.

8.6 Advantages of implementation of interoperability

Through of realisation of interoperability in the CAx chain it is possible for a global manufacturing enterprise to employ a "design anywhere, manufacture anywhere" approach in which conventional unidirectional information transfer is changed to the more complex possibilities afforded by bidirectional information transfer.

The advantages of such system include:

- The machining of legacy parts where there is no information available apart from the machine code.
- The machining of the same part with different CNC machines on the shop floor without needing to use any CAD/CAM software.
- The ability to increase production rates by using more CNC machines for manufacturing one part on the shop floor.

- The use of different machines in situations where the preferred machine has broken down without the need to redesign the code from the beginning of the CAPP chain.

- Improved interoperability in manufacturing by using standards to transfer information between CNC machines automatically.

8.7 Limitations of the XTSys

There are several limitations associated with the framework proposed in this research.

i. The ability to use parts with complex geometry for converting from one technology to another may prove to be difficult. Fully automatic comprehension still needs further development of the system to handle complex parts and to use a wider variety of technologies like EDM, laser cutting, etc.

ii. The prototype proposed in the thesis is not capable of handling data relating to geometric tolerances and multiple setups of fixtures. Correct handling of this information poses challenges that have to be resolved for further development of the framework.

iii. In this research the cutting tools selected for the operations have been chosen by the user. In the future it is expected that intelligent CNC machines which have knowledge of the machined geometry will be able to select or identify appropriate tools automatically.

9 Conclusions and future work

9.1 Introduction

In this chapter the conclusions that have been derived as the result of this research are provided. The contribution to knowledge provided by this research is highlighted together with suggestions for future work.

9.2 Conclusions

In the CNC manufacturing chain a lack of interoperability is a major challenge in achieving adaptability and flexibility in the CAD/CAM/CNC process chain.

A new approach for cross-technology interoperability based on semantic transformation has been proposed for CNC manufacturing of asymmetric rotational parts, which shows significant potential for seamless transfer of production from one machine type to another type.

Different standards have been developed to support universal interoperability between manufacturing resources. Information modelling in these standards is in many formats and various semantics differ from each specific resource to another. For supporting universal interoperability manufacturing chain a tailor-made standard is required.

The STEP-NC standard was developed to underpin adaptability between CAD/CAM/CNC systems, by proposing one format and one set of semantics to allow data transfer from one recourse to another. The research has shown that the semantics in STEP-NC, are however, technology specific and thus the gap for cross-technology interoperability still remains.

A new approach for cross-technology interoperability based on semantic transformation has been proposed for CNC manufacturing of asymmetric rotational parts, which shows significant potential for seamless transfer of production from one machine type to another type.

The use of the STEP-NC data model has been shown to represent the necessary part and machine information for semantic interpretation of a part process plan across different manufacturing technologies.

The performance capability of the experimental software reported in this thesis formed XTSys shows a strong case for the use of standards in interoperable systems. It is expected that such future systems based on standards will provide significant advantages in interoperability.

A prototype implementation of cross-technology interoperability has been successfully demonstrated for 3 asymmetric cylindrical parts proving interoperability across two different machine types.

9.3 Contributions to knowledge

The main contribution of this research to knowledge is a novel approach for achieving interoperability within shop floor between CNC machining centres with different technologies. This has been demonstrated using a prototype based on STEP-NC standards for enabling interoperability between milling and turn-mill machines.

9.4 Future work

During the course of this research a number of opportunities for taking the work further have been identified:

9.4.1 Extensibility to cover more operations

The proposed system covers milling operations such as `plane_milling`, `side_milling` and `bottom_and_side_milling` (which fits within

`Two5D_milling_operation` in ISO 14649-11), and drilling. Additional operations are identified as future work.

Similarly, in the turning technology, the system supports facing, contouring, grooving and drilling operations. Threading and knurling are identified as future work.

Adding these operations would require the XTSys manufacturing database to be updated with additional semantic transformation templates. Similar to those already existing for the current operations.

9.4.2 Communication with larger variety of technologies

The framework presented in this thesis can be integrated with a wider variety of technologies such as EDM, laser cutting, 3D printing, etc. and using other classifications of CNC machines. To achieve such a system, firstly the manufacturing technologies need to have a data model in the same format as the existing technologies, secondly the semantic transformation templates between other technologies needs to be established and coded into XTSys.

9.4.3 Integration of the STEP-NC Research

For achieving the universal cross-technology interoperability system for CNC manufacturing, extending the manufacturing lexicon in XTSys to support more technologies is necessary. For reaching this aim, adding more technologies and machine information to the STEP-NC standard would be necessary to serve as the basis for the additions required in the XTSys manufacturing database.

10 References

AGRAWAL, R., SHUKLA, S., KUMAR, S. & TIWARI, M. 2009. Multi-agent system for distributed computer-aided process planning problem in e-manufacturing environment. *The International Journal of Advanced Manufacturing Technology,* 44**,** 579-594.

ALTING, L. & ZHANG, H. 1989. Computer aided process planning: the state-of-the-art survey. *International Journal of Production Research,* 27**,** 553-585.

ALVARES, A., FERREIRA, J. & LORENZO, R. 2008. An integrated web-based CAD/CAPP/CAM system for the remote design and manufacture of feature-based cylindrical parts. *Journal of Intelligent Manufacturing,* 19**,** 643-659.

AMAITIK, S. & ENGIN KILIC, S. 2007. An intelligent process planning system for prismatic parts using STEP features. *The International Journal of Advanced Manufacturing Technology,* 31**,** 978-993.

ANON 2001. SYSTEMS ENGINEERING FUNDAMENTALS. *SYSTEMS ENGINEERING FUNDAMENTALS.*

ANWER, N. & CHEP, A. IPPA: a knowledge assisted process planning system. 2002. IEEE, 297-301.

AREZOO, B., RIDGWAY, K. & AL-AHMARI, A. 2000. Selection of cutting tools and conditions of machining operations using an expert system. *Computers in Industry,* 42**,** 43-58.

BENHABIB, B. 2003. *Manufacturing: design, production, automation and integration*, CRC Press.

BRANHAM, B. 2009. Selecting A VMC: Things To Consider. *Production Machining Magazine.*

CAY, F. & CHASSAPIS, C. 1997. An IT view on perspectives of computer aided process planning research. *Computers in Industry,* 34**,** 307-337.

CHOI, I., SUH, S., KIM, K., SONG, M., JANG, M. & LEE, B. 2006. Development process and data management of TurnSTEP: a STEP-compliant CNC system for turning. *International Journal of Computer Integrated Manufacturing,* 19**,** 546-558.

CHUNG, C. & PENG, Q. 2004. The selection of tools and machines on web-based manufacturing environments. *International Journal of Machine Tools and Manufacture,* 44**,** 317-326.

CUBICMACHINERY. 2013. *Cubic Machinery Inc.* [Online]. Available: http://cubicmachinery.com/yb/ [Accessed 28 March 2013 2013].

DMG. 2013. *DGM - technology for tomorrow* [Online]. Available: http://www.dmg.com [Accessed 26 June 2013].

DORF, R. & KUSIAK, A. 1994. *Handbook of design, manufacturing, and automation*, Wiley-Interscience.

ELMARAGHY, H., AGERMAN, E., DAVIES, B., ELMARAGHY, W., EVERSHEIM, W., HALEVI, D., HAM, I., JASPERSE, H., KALS, H. & NEE, A. 1993. Evolution and future perspectives of CAPP. *CIRP Annals-Manufacturing Technology,* 42**,** 739-751.

EMCO.MAIER.GES.M.B.H. 2007. *Hyperturn* [Online]. Available: http://www.emco-world.com.

ENGELKE, W. D. 1987. *How to integrate CAD/CAM systems: management and technology*, CRC PressI Llc.

EVERSHEIM, W. & SCHNEEWIND, J. 1993. Computer-aided process planning-State of the art and future development. *Robotics and Computer Integrated Manufacturing,* 10**,** 65-70.

EVERSHEIM, W. & SCHULZ, J. 1985. Survey of computer aided process planning systems. *Annals of the CIRP,* 34**,** 607-611.

EXECHON. 2013. Available: http://www.exechon.com.

FENG, S. & ZHANG, C. 1998. A modular architecture for rapid development of CAPP systems for agile manufacturing. *IIE transactions,* 30**,** 893-903.

GOLOGLU, C. 2004. A constraint-based operation sequencing for a knowledge-based process planning. *Journal of Intelligent Manufacturing,* 15**,** 463-470.

GRABOWIK, C. & KNOSALA, R. 2003. The method of knowledge representation for a CAPP system. *Journal of Materials Processing Technology,* 133**,** 90-98.

GROOVER, M. & ZIMMERS, E. 1983. *CAD/CAM: computer-aided design and manufacturing*, Prentice Hall.

GUNASEKARAN, A., MARTIKAINEN, T. & YLI-OLLI, P. 1995. Flexible manufacturing systems: An investigation for research and applications. *Manufacturing Research and Technology,* 23**,** 3-44.

GUPTA, T. & GHOSH, B. 1989. A survey of expert systems in manufacturing and process planning. *Computers in Industry,* 11**,** 195-204.

HAAS. 2013. *Hass Automation Inc* [Online]. Available: http://www.haascnc.com [Accessed 26 June 2013].

HAM, I. & LU, S. 1988. Computer-aided process planning: the present and the future. *CIRP Annals-Manufacturing Technology,* 37**,** 591-601.

HARDINGE. 2013. Available: http://www.hardinge.com.

HARDWICK, M. 2002. *Digital Manufacturing Using STEP-NC*, Society of Manufacturing Engineers.

HARDWICK, M. & LOFFREDO, D. 2006. Lessons learned implementing STEP-NC AP-238. *International Journal of Computer Integrated Manufacturing,* 19**,** 523-532.

HECHT, H. Systems engineering for software-intensive projects. Application-Specific Systems and Software Engineering and Technology, 1999. ASSET '99. Proceedings. 1999 IEEE Symposium on, 1999 1999. 58-61.

HEUSINGER, S., ROSSO, R., KLEMM, P., NEWMAN, S. & RAHIMIFARD, S. 2006. Integrating the CAx process chain for STEP-compliant NC manufacturing of asymmetric parts. *International Journal of Computer Integrated Manufacturing,* 19**,** 533-545.

HNK. 2013. *HNK MACHINE TOOL CO., LTD.* [Online]. Available: http://www.hnkkorea.com [Accessed 2 June 2013].

HOLLAND, J. H. 1992. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, the MIT Press.

HOLLAND, M. 1989. *When the Machine Stopped: A Cautionary Tale from Industrial America, Boston*.

HOU, M. 2008. CAD/CAM integration based on machining features for prismatic parts.

HU, C., LI, Z., ZHENG, L., LI, N. & WEN, P. 2008. An XML-based implementation of manufacturing route sheet documents for context-sensitive and web-based process planning. *International Journal of Computer Integrated Manufacturing,* 21**,** 647-656.

INDEX. 2013. *INDEX-Werke GmbH & Co. KG Hahn & Tessky* [Online]. Available: http://www.index-werke.de [Accessed 28 June 2013].

ISO 6983-1 1982. Industrial automation systems and integration - product data representation and exchange. International Standards Organisation.

ISO 10303-11 1994. *Industrial automation systems and integration - product data representation and exchange. part 11: Description methods: Express language reference manual.* International Standards Organisation.

ISO 10303-238 2006. *Industrial automation systems and integration - product data representation and exchange. part 238: Application protocol: Application interpreted model for computerized numerical controllers.* International Standards Organisation.

ISO 14649-1 2002. *Industrial automation systems and integration - physical device control - data model for computerized numerical controllers. part 1: Overview and fundamental principles.*: International Standards Organisation.

ISO 14649-10 2002. *Industrial automation systems and integration - physical device control - data model for computerized numerical controllers. part 10: General process data.* International Standards Organisation.

ISO 14649-11 2002. *Industrial automation systems and integration - physical device control - data model for computerized numerical controllers. part 11: Process data for milling.* International Standards Organisation.

ISO 14649-12 2004. Industrial automation systems and integration - physical device control - data model for computerized numerical controllers. *part 12: Process data for turning.* International Standards Organisation.

ISO 14649-111 2002. *Industrial automation systems and integration - physical device control - data model for computerized numerical controllers. part 111: Tools for milling.* International Standards Organisation.

ISO 14649-121 2004. Industrial automation systems and integration - physical device control - data model for computerized numerical controllers. *part 121: Tools for turning.* International Standards Organisation.

ISO 14649-201 2011. Industrial automation systems and integration - physical device control - data model for computerized numerical controllers. *part 201: Machine tool data for cutting process.* International Standards Organisation.

ISO-TC184-CS4-WG3-T24 2009. International STEP-NC Demonstration, Bath, UK 2009. STEP NC.

ISO/IEC 7498-1 1994. *Information technology -- Open Systems Interconnection -- Basic Reference Model*. *The basic model.* International Standards Organisation.

JIANG, B., LAU, H., CHAN, F. & JIANG, H. 1999. An automatic process planning system for the quick generation of manufacturing process plans directly from CAD drawings. *Journal of Materials Processing Technology,* 87**,** 97-106.

KALPAKJIAN, S. 2014. *Manufacturing engineering and technology,* Jurong, Singapore, Jurong, Singapore : Pearson Education South Asia.

KALPAKJIAN, S. & SCHMID, S. R. 1992. Manufacturing engineering and technology. *Reading, Massachusetts*.

KAMRANI, A., SFERRO, P. & HANDELMAN, J. 1995. Critical issues in design and evaluation of computer aided process planing systems. *Computers & Industrial Engineering,* 29**,** 619-623.

KENTCNC. 2013. Available: http://www.kentcnc.com [Accessed 26 June 2013].

KIRITSIS, D. 1995. A review of knowledge-based expert systems for process planning. Methods and problems. *The International Journal of Advanced Manufacturing Technology,* 10**,** 240-262.

KOCHAN, D. 1986. *CAM Developments in computer-integrated manufacturing*.

KRYSSANOV, V., KLESHCHEV, A., FUKUDA, Y. & KONISHI, K. 1998. Building a logical model in the machining domain for CAPP expert systems. *International Journal of Production Research,* 36**,** 1075-1089.

KRZIC, P., STOIC, A. & KOPAC, J. 2009. STEP-NC: A new programming code for the CNC machines. *Strojniški vestnik,* 55**,** 406-417.

KUMAR, S., NASSEHI, A., NEWMAN, S., ALLEN, R. & TIWARI, M. 2007. Process control in CNC manufacturing for discrete components: A STEP-NC compliant framework. *Robotics and Computer Integrated Manufacturing,* 23**,** 667-676.

LEE, W., BANG, Y., RYOU, M., KWON, W. & JEE, H. 2006. Development of a PC-based milling machine operated by STEP-NC in XML format. *International Journal of Computer Integrated Manufacturing,* 19**,** 593-602.

LEUNG, H. 1996. Annotated bibliography on computer-aided process planning. *The International Journal of Advanced Manufacturing Technology,* 12**,** 309-329.

LIAO, S. 2005. Expert system methodologies and applications--a decade review from 1995 to 2004. *Expert Systems with Applications,* 28**,** 93-103.

LIU, R., ZHANG, C., NASSEHI, A. & NEWMAN, S. A STEP-NC programming system for prismatic parts. 12th International Manufacturing Conference, 2006. Aedermannsdorf, Switzerland: Trans Tech Publications, 1984-, 1108-1111.

LIU, Y., GUO, X., LI, W., YAMAZAKI, K., KASHIHARA, K. & FUJISHIMA, M. 2007. An intelligent NC program processor for CNC system of machine tool. *Robot. Comput.-Integr. Manuf.,* 23**,** 160-169.

LU, S., YU, T. & LIU, T. Research on CAPP Technology Oriented to CNC. Control, Automation, Robotics and Vision, 2006. 1-5.

MARRI, H., GUNASEKARAN, A. & GRIEVE, R. 1998. Computer-aided process planning: a state of art. *The International Journal of Advanced Manufacturing Technology,* 14**,** 261-268.

MAS.TECH. 2013. *Mas Tech* [Online]. Available: http://www.mastech-strojevi.com/ [Accessed 11 March 2012 2012].

MAZAK. 2013. *Yamazaki Mazak Corporation* [Online]. Available: http://www.mazak.eu [Accessed 5 June 2013].

MCMAHON, C. & BROWNE, J. 1993. *CADCAM: from principles to practice*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

METAXIOTIS, K., ASKOUNIS, D. & PSARRAS, J. 2002. Expert systems in production planning and scheduling: A state-of-the-art survey. *Journal of Intelligent Manufacturing,* 13**,** 253-260.

MEYERS, A. & SLATTERY, T. 2001. *Basic machining reference handbook*, Industrial Press Inc.

NASSEHI, A. 2007. *The realisation of CAD/CAM/CNC interoperability in prismatic part manufacturing.* Thesis (Ph.D.) - University of Bath, 2007.

NASSEHI, A., LIU, R. & NEWMAN, S. T. 2007a. A new software platform to support feature-based process planning for interoperable STEP-NC manufacture. *International Journal of Computer Integrated Manufacturing,* 20**,** 669-683.

NASSEHI, A., NEWMAN, S., XU, X. & ROSSO, R. 2007b. Toward interoperable CNC manufacturing. *International Journal of Computer Integrated Manufacturing,* 21**,** 222-230.

NEWMAN, S., NASSEHI, A., XU, X., ROSSO, R., WANG, L., YUSOF, Y., ALI, L., LIU, R., ZHENG, L. & KUMAR, S. 2008. Strategic advantages of interoperability for global manufacturing using CNC technology. *Robotics and Computer Integrated Manufacturing,* 24**,** 699-708.

NEWMAN, S., T, ALLEN, R. & ROSSO, R. 2003. CAD/CAM solutions for STEP-compliant CNC manufacture. *International Journal of Computer Integrated Manufacturing,* 16**,** 590-597.

NEWMAN, S., T & NASSEHI, A. 2007. Universal manufacturing platform for CNC machining. *CIRP Annals-Manufacturing Technology,* 56**,** 459-462.

NIEBEL, B. 1965. Mechanized process selection for planning new designs. *ASME paper,* 737.

OKUMA. 2013. *Okuma Australia Pty Ltd* [Online]. Available: http://www.okumaaustralia.com.au/ 2013].

PARKS, C. IGES as an interchange format for integrated circuit design. Design Automation, 1984 New Mexico. IEEE Press, 273-274.

PARSONS, J. & STULEN, F. 1958. *Motor controlled apparatus for positioning machine tools*. United States patent application.

PENG, G., LIU, W. & ZHANG, X. 2005. An Internet-based system for setup planning in machining operations. *Engineering of Complex Computer Systems.* Shanghai-China.

PENG, Q., HALL, F. & LISTER, P. 2000. Application and evaluation of VR-based CAPP system. *Journal of Materials Processing Technology,* 107**,** 153-159.

PHAM, D. T. & GOLOGLU, C. 2001. A computer aided process planning system for concurrent engineering. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 215**,** 1117-1131.

PRATT, M. J. Extension of ISO 10303, the STEP standard, for the exchange of procedural shape models. Shape Modeling Applications, 2004. Proceedings, 7-9 June 2004 2004. 317-326.

PRITSCHOW, G., ALTINTAS, Y., JOVANE, F., KOREN, Y., MITSUISHI, M., TAKATA, S., VAN BRUSSEL, H., WECK, M. & YAMAZAKI, K. 2001. Open controller architecture–past, present and future. *CIRP Annals-Manufacturing Technology,* 50**,** 463-470.

QIU, Z., CHEN, Y., ZHOU, Z., ONG, S. & NEE, A. 2001. Multi-user NC machining simulation over the WWW. *The International Journal of Advanced Manufacturing Technology,* 18**,** 1-6.

QUASER. 2013. *Quaser Machine Tools Inc.* [Online]. Available: http://www.quaser.com [Accessed 26 June 2013].

ROCHA, J., RAMOS, C. & VALE, Z. Process planning using a genetic algorithm approach. Assembly and Task Planning, 1999. (ISATP '99) Proceedings of the 1999 IEEE International Symposium on, 1999 1999. 338-343.

ROSSO, R., NEWMAN, S. & RAHIMIFARD, S. 2004. The adoption of STEP-NC for the manufacture of asymmetric rotational components. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* 218**,** 1639-1644.

SARAVI, M., NEWNES, L., MILEHAM, A. & GOH, Y. 2008. Estimating Cost at the Conceptual Design Stage to Optimize Design in terms of Performance and Cost. *In:* CURRAN, R., CHOU, S.-Y. & TRAPPEY, A. (eds.) *Collaborative Product and Service Life Cycle Management for a Sustainable World.* Springer London.

SCHROEDER, C. 1998. *Printed circuit board design using AutoCAD*, Butterworth-Heinemann.

SCHROEDER, T. & HOFFMANN, M. 2006. Flexible automatic converting of NC programs. A cross-compiler for structured text. *International Journal of Production Research,* 44**,** 2671-2679.

SHAH, J., SREEVALSAN, P. & MATHEW, A. 1991. Survey of CAD/feature-based process planning and NC programming techniques. *Computer-Aided Engineering,* 8**,** 25-33.

SHEN, W., WANG, L. & HAO, Q. 2006. Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* 36**,** 563-577.

SHIMAMURA, A., MORIYAMA, M. & KASUGA, K. 1996. Data detection method for curve and surface data in STEP data exchange system. *The Japan Society for Precision Engineering,* 62**,** 701-705.

SHIN, S., SUH, S. & STROUD, I. 2007. Reincarnation of G-code based part programs into STEP-NC for turning applications. *Computer-Aided Design,* 39**,** 1-16.

SHU-HSIEN, L. 2005. Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert Systems with Applications,* 28**,** 93-103.

SHUNMUGAM, M., BHASKARA REDDY, S. & NARENDRAN, T. 2000. Selection of optimal conditions in multi-pass face-milling using a genetic algorithm. *International Journal of Machine Tools and Manufacture,* 40**,** 401-414.

SHUNMUGAM, M., MAHESH, P. & BHASKARA REDDY, S. 2002. A method of preliminary planning for rotational components with C-axis features using genetic algorithm. *Computers in Industry,* 48**,** 199-217.

SRINIVASAN, V. & FISCHER, G. 1996. Direct interface integration of CAD and CAM software—a milling application. *Journal of materials processing technology,* 61**,** 93-98.

STAMA. 2013. *ATAMA Engineering* [Online]. Available: www.stama.de [Accessed 26 June 2013].

STENERSON, J. & CURRAN, K. 2005. *Computer numerical control: operation and programming*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

STEPHENSON, D. & AGAPIOU, J. 2006. *Metal cutting theory and practice*, CRC Press.

STEUDEL, H. 1984. Computer-aided process planning: past, present and future. *International Journal of Production Research,* 22**,** 253-266.

STRASSER, W. & SEIDEL, H.-P. 1989. *Theory and practice of geometric modeling*, Springer-Verlag.

SUH, S. & CHEON, S. 2002. A framework for an intelligent CNC and data model. *The International Journal of Advanced Manufacturing Technology,* 19**,** 727-735.

SUH, S., CHO, J. & HONG, H. 2002. On the architecture of intelligent STEP-compliant CNC. *International Journal of Computer Integrated Manufacturing,* 15**,** 168-177.

SUH, S., CHUNG, D., LEE, B., SHIN, S., CHOI, I. & KIM, K. 2006. STEP-compliant CNC system for turning: Data model, architecture, and implementation. *Computer aided design,* 38**,** 677-688.

SUH, S., KANG, S., CHUNG, D. & STROUD, I. 2008. *Theory and design of CNC systems*, Springer Verlag.

SUH, S., LEE, B., CHUNG, D. & CHEON, S. 2003. Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. *Computer-Aided Design,* 35**,** 1069-1083.

SUN, J., ZHANG, Y. & NEE, A. 2001. A distributed multi-agent environment for product design and manufacturing planning. *International Journal of Production Research,* 39**,** 625-645.

TALAVAGE, J. & HANNAM, R. 1988. *Flexible manufacturing systems in practice: applications, design, and simulation*, CRC.

TAVAKOLI, A.-B. 1993. *Feature based workshop oriented NC planning for asymmetric rotational parts.* PhD Thesis. Department of Manufacturing Engineering. Loughborough University.

TOLIO, T., COBIANCHI, P., MARIA & MORIGGI, P. 2010. *METHOD FOR AUTOMATICALLY PARTITIONING A PART PROGRAM INTO FUNDAMENTAL OPERATIONS*. WO/2010/116265.

TSATSOULIS, C. & KASHYAP, R. L. 1988. A system for knowledge-based process planning. *Artificial Intelligence in Engineering,* 3**,** 61-75.

TWIGG, D., VOSS, C. A. & WINCH, G. M. 1992. Implementing integrating technologies: developing managerial integration for CAD/CAM. *International Journal of Operations & Production Management,* 12**,** 76-91.

VICHARE, P. 2009. *A Novel Methodology for Modelling CNC Machining System Resources.* Doctor of Philosophy, University of Bath.

WAIYAGAN, K. & BOHEZ, E. L. 2009. Intelligent feature based process planning for five-axis mill-turn parts. *Computers in Industry,* 60**,** 296-316.

WANG, L., BALASUBRAMANIAN, S., NORRIE, D. & BRENNAN, R. Agent-based control system for next generation manufacturing. Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings, 1998. IEEE, 78-83.

WANG, L. & BI, Z. 2013. Challenges for Better Sustainable Manufacturing. *Advances in Sustainable and Competitive Manufacturing Systems.* Springer.

WANG, L., JIN, W. & FENG, H.-Y. 2006. Embedding machining features in function blocks for distributed process planning. *International Journal of Computer Integrated Manufacturing,* 19**,** 443-452.

WANG, L., SAMS, R., VERNER, M. & XI, F. 2003. Integrating Java 3D model and sensor data for remote monitoring and control. *Robotics and Computer-Integrated Manufacturing,* 19**,** 13-19.

WERNER DANKWORT, C., WEIDLICH, R., GUENTHER, B. & BLAUROCK, J. E. 2004. Engineers' CAx education—it's not only CAD. *Computer-Aided Design,* 36**,** 1439-1450.

XU, X. 2006. Realization of STEP-NC enabled machining. *Robotics and Computer Integrated Manufacturing,* 22**,** 144-153.

XU, X. 2009. *Integrating advanced computer-aided design, manufacturing, and numerical control: principles and implementations*, Information Science Reference.

XU, X., KLEMM, P., PROCTOR, F. & SUH, S. 2006. STEP-compliant process planning and manufacturing. *International Journal of Computer Integrated Manufacturing,* 19**,** 491-494.

XU, X. & NEWMAN, S. 2006. Making CNC machine tools more open, interoperable and intelligent: a review of the technologies. *Computers in Industry,* 57**,** 141-152.

XU, X., WANG, H., MAO, J., NEWMAN, S., KRAMER, T., PROCTOR, F. & MICHALOSKI, J. 2005. STEP-compliant NC research: the search for intelligent CAD. CAPP/CAM/CNC integrationInternational Journal of Production Research, 2005, 43 (17): 3703-3743.

XU, X. & WANG, J. Development of a G-code free, STEP-compliant CNC lathe. International Mechanical Engineering Congress and Exposition, 2004 Anaheim, CA, USA. 1-5.

XU, X., WANG, L. & NEWMAN, S. 2011. Computer-aided process planning-A critical review of recent developments and future trends. *International Journal of Computer Integrated Manufacturing,* 24**,** 1-31.

YOUNIS, M. & ABDEL WAHAB, M. 1997. A CAPP expert system for rotational components. *Computers & Industrial Engineering,* 33**,** 509-512.

YUE, Y., DING, L., AHMET, K., PAINTER, J. & WALTERS, M. 2002. Study of neural network techniques for computer integrated manufacturing. *Engineering Computations,* 19**,** 136-157.

YUSOF, Y., NEWMAN, S., NASSEHI, A. & CASE, K. Interoperable CNC system for turning operations. World Academy of Science, Engineering and Technology 2009. WASET, 941-947.

YUSOF, Y. & NOR, M. F. M. 2012. Review of STEP-NC Compliant System for Turn-Mill Operations. *Advanced Materials Research,* 383**,** 2827-2831.

YUSOF, Y., ROSSO JR, R., CASE, K. & NEWMAN, S. The design of a STEP-NC compliant CAD/CAP/CAM system for the manufacture of rotational parts on a CNC turning centre, in'Innovations in Manufacturing'. 23rd International Manufacturing Conference, 2006 Northern Ireland. University of Bath.

ZHANG, W. & XIE, S. 2007. Agent technology for collaborative process planning: a review. *The International Journal of Advanced Manufacturing Technology,* 32**,** 315-325.

ZHANG, X. 2012. *Process comprehension for interoperable CNC manufacturing.* Thesis (Ph.D.) - University of Bath, 2012.

ZHANG, X., LIU, R., NASSEHI, A. & NEWMAN, S. 2010. A STEP-compliant process planning system for CNC turning operations. *Robotics and Computer-Integrated Manufacturing*.

ZHANG, X., NASSEHI, A., DHOKIA, V. G. & NEWMAN, S. T. 2012. Refining Process Logic From CNC Part Programmes for Integrated STEP-NC Compliant Manufacturing of Prismatic Parts. *In:* ELMARAGHY, H. A. (ed.) *Enabling Manufacturing Competitiveness and Economic Sustainability.* Springer Berlin Heidelberg.

ZHANG, X., NASSEHI, A., SAFAIEH, M. & NEWMAN, S. T. 2013. Process comprehension for shopfloor manufacturing knowledge reuse. *International Journal of Production Research***,** 1-15.

ZHAO, Y., RIDGWAY, K. & AL-AHMARI, A. 2002. Integration of CAD and a cutting tool selection system. *Computers & Industrial Engineering,* 42**,** 17-34.

ZHOU, B., HE, Z.-J. & TANG, J.-F. An adaptive model of virtual enterprise based on dynamic web service composition. Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on, 2005. IEEE, 284-289.

11 Appendix A. Publications

The author published a number of scientific peer reviewed papers during the research that documented in this thesis.

11.1 Journal articles

1. **Mehrdad Safaieh**, Aydin Nassehi, and Stephen T. Newman. "A novel methodology for cross-technology interoperability in CNC machining" *Robotics and Computer-Integrated Manufacturing, Volume 29, Issue 3, June 2013, Pages 79-87*

In CNC part programmes, the lack of standardisation for representing part geometry and semantics of manufacturing operations leads to the necessity for existence of a unique part programme for each machine. Generating multiple programmes for producing the same part is not a value adding activity and is very time consuming. This wasteful activity can be eliminated if users are given the ability to write an NC program for a specific machine and robustly convert the program to syntax suitable for another CNC machine with a different structure. This, cross-technology interoperability, would enable for parts manufactured on old CNC machines using legacy code to be manufactured on new CNC machines by automatically converting the programmes. Every NC programme is written based on various categories of information such as: cutting tool specifications, process planning knowledge and machine tool information. This paper presents an approach for cross-technology interoperability by refining high-level process information (i.e., geometric features on the part and embedded manufacturing resource data) from NC programmes. These refined items of information stored in compliance with the ISO14649 (STEP-NC) standard may then be combined with new manufacturing resource information to generate NC code in a format that is compatible with machines based on different technologies. The authors provide a framework for this process of identification, semantic interpretation and re-integration of information. The focus of this paper is on asymmetric rotational components as the initial application area. To demonstrate the proposed cross-technology interoperability approach, a C-axis CNC turn-mill machine and a 4 axis CNC machining centre have been used with a simple test component.

2. Xianzhi Zhang , Aydin Nassehi , **Mehrdad Safaieh** & Stephen T. Newman, "Process comprehension for shopfloor manufacturing knowledge reuse", *International Journal of Production Research*, *DOI:10.1080/00207543.2012.757669*

Computer numerical controlled (CNC) machines play an important role in the production capacity of manufacturing enterprises. With the advance of computing technology, computer-aided systems (CAx) have been intensively used together with the CNC machines. The information flow from CAx to CNC machines is unidirectional, due to the wide- spread use of G&M codes to programme the CNC machines and the mechanism of the generation of part programmes. The CNC machines at the shopfloor have been isolated from the CAx chain. There is no automatic way to capture and feedback the shopfloor knowledge. Reusing shopfloor process knowledge offers the enterprises opportunities to improve manufacturing quality and control while enabling savings in cost and time. Rapid product development relies heavily on quick and reliable process planning and knowledge reuse to facilitate the process plan efficiently and effectively. In this research, the process comprehension approach has been utilised to capture the process knowledge at the shopfloor. A novel method has been proposed to reuse the process knowledge with different manufacturing resources. In this paper, a short review on manufacturing knowledge management is provided. The process comprehension approach is then pre- sented. An example part is used as the case study to illustrate the knowledge capture using process comprehension and how the process knowledge can be utilised to manufacture the product with new manufacturing resources.

11.2 Conference papers

1. **M. Safaieh**, A. Nassehi, S.T. Newman, "Realization of interoperability between a C-axis CNC turn-mill centre and 4-axis CNC machining centre", *Proceeding of the 21th International Conference on Flexible Automation and Intelligent Manufacturing Conference (FAIM2011), Taiwan, 26th-29th June 2011, pp37-43.*

2. **M. Safaieh**, A. Nassehi, S.T. Newman, "Cross Technology Interoperability for CNC Metal Cutting Machines", Proceeding of the 21th International Conference on Production Research (ICPR), Stuttgart, Germany, 31th-4th August 2011.

3. **M. Safaieh**, A. Nassehi, S.T. Newman, "Adapting STEP-NC programs for interoperability between different CNC technologies", *Proceeding of the 37th International MATADOR Conference, Manchester, UK, 25th-27th July 2012, pp45-48.*

12 Appendix B. Program Listing for the test parts

12.1 G&M Code for Dugard listing

```
%
G1900 D50 L100 K0.5
G17 G21 G90 G94 G57

T11 D11 M6
S3000 F1000 M3
G0 G43 X50 Y50 Z100 H11
M8

G0 X-60 Y0
Z10
G1 Z-1.5
X60
Z-2
X-60
Z10
M9

T10 D10 M6
S4000 F800 M3
G0 G43 X50 Y50 Z100 H10
M8

G0 X-25 Y21
Z10
G1 Z-3
X21
Y-21
X-21
Y30

G1 X-25 Y21
G1 Z-5
X21
Y-21
X-21
Y30

G1 X-25 Y21
G1 Z-6.5
X21
Y-21
X-21
Y30
G1 X-25 Y21
G1 Z-7
X21
Y-21
X-21
Y30
Z10
M9

T13 D13 M6
S5000 F800 M3
G0 G43 X50 Y50 Z100 H13
M8

G0 X-35 Y0
Z10
G1 Z-8.5
X-24.21
G2 X24.21 Y0 I24.21 J0
G2 X-24.21 Y0 I-24.21 J0

G1 Z-10
G2 X24.21 Y0 I24.21 J0
G2 X-24.21 Y0 I-24.21 J0

G1 Z-11.5
G2 X24.21 Y0 I24.21 J0
G2 X-24.21 Y0 I-24.21 J0

G1 Z-12
G2 X24.21 Y0 I24.21 J0
G2 X-24.21 Y0 I-24.21 J0
G2 X24.21 Y0 I24.21 J0
G1 Z10
S5001
G0 X-40 Y0
G1 Z-3.5
X40
Z-4.5
X-40
Z10
G0 Z100

T5 D5 M6
```

```
S3000 F800 M3
G0 G43 X50 Y50 Z100 H5
M8
G0 X-7.5 Y0 Z1
G1 Z-7
G2 X7.5 Y0 I7.5 J0
G2 X-7.5 Y0 I-7.5 J0
G1 X-2.5 Y0
G2 X2.5 Y0 I2.5 J0
G2 X-2.5 Y0 I-2.5 J0
M9

M5
M30
%
```

12.2 G&M Codes for Hyundai-Kia listing

```
%
O0066
G1900 D50 L25 K0.5
G21
G10 P0 X0 Z-70

(FACING)
G28 U0
T0707
G50 S2000
G96 S200 M3
G00 G40 G99 X55 Z0 M8
G01 X-2 F0.25
G00 X55 Z2 M9

(TURNING)

G71 U1 R1
G71 P100 Q200 U0.3 W0.1 F0.25
N100 G00 X40 Z2
G01 G42 Z0 F0.15
G01 X42.42
G01 Z-10
N200 G40 X55 Z2

G70 P100 Q200

G28 U0 W0


(MILLING FOR INTERNAL HOLE)

G28 U0
T0606
M5
M43
G97 S1500 M13
G00 G40 G98 X0 Z2 M8
G83 Z-7 Q200 F100
G80
M15
M40

(MIILING THE SQUARE)
M5
M43
M111
G28 H0 W0
G97 S1500 M13
G00 G40 G98 G54 X60 Z10 M8
G01 Z-5 F2500
G112
G01 C-15
G41 X30 F100
X-30 C-15
X-30 C15
X30 C15
C-15
G40
G113
G00 Z100 M9
M15
M110
M40


(MILLING THE SLOT)

G28 U0 W0
T1111
M5
M43
M111
G28 H0
G97 S1500 M13
G00 G40 G98 G54 X60 Z10 M8
G01 Z-3 F2500
C0
G01 X0
X60
C180
X0
X60
G28 U0 W0

M15
M110
M40
M2
%
```

12.3 STEP-NC code for Milling machining centre part 1

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('GENERATED ISO 14649-11 FILE','AUTOMATIC OUTPUT OF
UPCi FROM A CNC PART PROGRAMME'), '1');
FILE_NAME('EXAMPLE.STP', '2013-04-17', ('MEHRDAD SAFAIEH'),
('UNIVERSITY OF BATH, BATH,UK'),$,'ISO 14649',$);
FILE_SCHEMA(('MACHINING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;

DATA;
#1=PROJECT('RECOGNISED ISO 14649 PART 21  FILE FROM G&M
CODES',#2,(#3),$,$,$);
#2=WORKPLAN('MAIN WORKPLAN',(#4,#5,#6,#7,#8),$,#9,$);
#3=WORKPIECE('WORKPIECE50.0X50.0X60.0',$,$,$,$,#21,());
#4=MACHINING_WORKINGSTEP('WS PLANAR_FACE1',#10,#11,#12,$);
#5=MACHINING_WORKINGSTEP('WS PLANAR_FACE2',#10,#44,#45,$);
#6=MACHINING_WORKINGSTEP('WS PLANAR_FACE3',#10,#82,#83,$);
#7=MACHINING_WORKINGSTEP('WS SLOT1',#10,#122,#123,$);
#8=MACHINING_WORKINGSTEP('WS POCKET1',#10,#152,#153,$);
#9=SETUP('SETUP',#179,#10,(#180));
#10=ELEMENTARY_SURFACE('SECURITY PLANE',#13);
#11=PLANAR_FACE('PLANAR_FACE1',#3,(#12),#17,#18,#19,#20,$,());
#12=PLANE_ROUGH_MILLING($,$,'PLANAR_FACE1',$,$,#26,#27,#28,$,#29,#30
,#31,$,$);
#13=AXIS2_PLACEMENT_3D('SECURITY PLANE PLACEMENT',#14,#15,#16);
#14=CARTESIAN_POINT('SECURITY PLANE: LOCATION',(-25.0,-25.0,-
15.0,0.0,0.0));
#15=DIRECTION('AXIS',(0.0,0.0,1.0));
#16=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#17=AXIS2_PLACEMENT_3D('PLANAR_FACE1 PLACEMENT',#34,#35,#36);
#18=ELEMENTARY_SURFACE('PLANAR_FACE1 DEPTH PLANE',#37);
#19=LINEAR_PATH($,#41,#42);
#20=LINEAR_PROFILE($,#43);
#21=BLOCK('WORKPIECE BLOCK',#22,50.0,50.0,-60.0);
#22=AXIS2_PLACEMENT_3D('WORKPIECE BLOCK PLACEMENT',#23,#24,#25);
#23=CARTESIAN_POINT('WORKPIECE BLOCK: LOCATION',(0.0,0.0,0.0));
#24=DIRECTION('AXIS',(0.0,0.0,1.0));
#25=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#26=MILLING_CUTTING_TOOL('T11',#32,(),$,$,$);
#27=MILLING_TECHNOLOGY(0.016,.TCP.,$,50.0,$,$,$,$,$);
#28=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#29=PLUNGE_RAMP($,$);
#30=PLUNGE_RAMP($,$);
#31=UNIDIRECTIONAL($,$,$,$);
#32=FACEMILL(#33,$,$,$,$);
#33=MILLING_TOOL_DIMENSION(60.0,$,$,$,0.0,$,$);
#34=CARTESIAN_POINT('PLANAR_FACE1',(-35.0,25.0,0.0));
```

```
#35=DIRECTION('AXIS',(0.0,0.0,1.0));
#36=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#37=AXIS2_PLACEMENT_3D('PLANAR_FACE1 DEPTH',#38,#39,#40);
#38=CARTESIAN_POINT('PLANAR_FACE1 DEPTH',(120.0,0.0,-2.0));
#39=DIRECTION('AXIS',(0.0,0.0,1.0));
#40=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#41=TOLERANCED_LENGTH_MEASURE(180.0,$);
#42=DIRECTION('PLANAR FACE DIRECTION',(0.0,1.0,0.0));
#43=NUMERIC_PARAMETER('null',60.0,'mm');
#44=PLANAR_FACE('PLANAR_FACE2',#3,(#45),#46,#47,#48,#49,$,(#50));
#45=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'PLANAR_FACE2',$,$,#51,#52,#53
,$,#54,#55,#56,$,$,$,$);
#46=AXIS2_PLACEMENT_3D('PLANAR_FACE2 PLACEMENT',#59,#60,#61);
#47=ELEMENTARY_SURFACE('PLANAR_FACE2 DEPTH PLANE',#62);
#48=LINEAR_PATH($,#66,#67);
#49=LINEAR_PROFILE($,#68);
#50=BOSS('PLANAR_FACE2_BOSS',#3,(#45),#69,#47,#70,$);
#51=MILLING_CUTTING_TOOL('T10',#57,(),$,$,$);
#52=MILLING_TECHNOLOGY(0.013,.TCP.,$,66.666,$,$,$,$,$);
#53=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#54=PLUNGE_RAMP($,$);
#55=PLUNGE_RAMP($,$);
#56=CONTOUR_PARALLEL($,$,$,$);
#57=ENDMILL(#58,$,$,$,$);
#58=MILLING_TOOL_DIMENSION(12.0,$,$,$,0.0,$,$);
#59=CARTESIAN_POINT('PLANAR_FACE2',(4.0,55.0,0.0));
#60=DIRECTION('AXIS',(0.0,0.0,1.0));
#61=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#62=AXIS2_PLACEMENT_3D('PLANAR_FACE2 DEPTH',#63,#64,#65);
#63=CARTESIAN_POINT('PLANAR_FACE2 DEPTH',(-4.0,-9.0,-7.0));
#64=DIRECTION('AXIS',(0.0,0.0,1.0));
#65=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#66=TOLERANCED_LENGTH_MEASURE(74.289,$);
#67=DIRECTION('PLANAR FACE DIRECTION',(0.0,1.0,0.0));
#68=NUMERIC_PARAMETER('null',12.0,'mm');
#69=AXIS2_PLACEMENT_3D('PLANAR_FACE2 PLACEMENT',#71,#72,#73);
#70=GENERAL_CLOSED_PROFILE($,#74);
#71=CARTESIAN_POINT('LOCATIONPLANAR_FACE2',(0.0,0.0,0.0));
#72=DIRECTION('AXIS',(0.0,0.0,1.0));
#73=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#74=COMPOSITE_CURVE('BOUNDARY: PLANAR_FACE2',(#75),.F.);
#75=COMPOSITE_CURVE_SEGMENT(.CONTINUOUS.,.T.,#76);
#76=POLYLINE('POLYLINE FOR CONTOUR:
PLANAR_FACE2',(#77,#78,#79,#80,#81));
#77=CARTESIAN_POINT('POLYLINE POINT 1',(36.0,-45.0,0.0));
#78=CARTESIAN_POINT('POLYLINE POINT 2',(6.0,-45.0,0.0));
#79=CARTESIAN_POINT('POLYLINE POINT 3',(6.0,-15.0,0.0));
#80=CARTESIAN_POINT('POLYLINE POINT 4',(36.0,-15.0,0.0));
```

```
#81=CARTESIAN_POINT('POLYLINE POINT 5',(36.0,-45.0,0.0));
#82=PLANAR_FACE('PLANAR_FACE3',#3,(#83),#84,#85,#86,#87,$,(#88));
#83=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'PLANAR_FACE3',$,$,#89,#90,#91
,$,#92,#93,#94,$,$,$,$);
#84=AXIS2_PLACEMENT_3D('PLANAR_FACE3 PLACEMENT',#97,#98,#99);
#85=ELEMENTARY_SURFACE('PLANAR_FACE3 DEPTH PLANE',#100);
#86=LINEAR_PATH($,#104,#105);
#87=LINEAR_PROFILE($,#106);
#88=BOSS('PLANAR_FACE3_BOSS',#3,(#83),#107,#85,#108,$);
#89=MILLING_CUTTING_TOOL('T13',#95,(),$,$,$);
#90=MILLING_TECHNOLOGY(0.0133,.TCP.,$,83.333,$,$,$,$,$);
#91=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#92=PLUNGE_RAMP($,$);
#93=PLUNGE_RAMP($,$);
#94=CONTOUR_PARALLEL($,$,$,$);
#95=ENDMILL(#96,$,$,$,$);
#96=MILLING_TOOL_DIMENSION(6.0,$,$,$,0.0,$,$);
#97=CARTESIAN_POINT('PLANAR_FACE3',(49.0,25.0,0.0));
#98=DIRECTION('AXIS',(0.0,0.0,1.0));
#99=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#100=AXIS2_PLACEMENT_3D('PLANAR_FACE3 DEPTH',#101,#102,#103);
#101=CARTESIAN_POINT('PLANAR_FACE3 DEPTH',(-48.21,0.0,-12.0));
#102=DIRECTION('AXIS',(0.0,0.0,1.0));
#103=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#104=TOLERANCED_LENGTH_MEASURE(65.21,$);
#105=DIRECTION('PLANAR FACE DIRECTION',(0.0,1.0,0.0));
#106=NUMERIC_PARAMETER('null',6.0,'mm');
#107=AXIS2_PLACEMENT_3D('PLANAR_FACE3 PLACEMENT',#109,#110,#111);
#108=GENERAL_CLOSED_PROFILE($,#112);
#109=CARTESIAN_POINT('LOCATIONPLANAR_FACE3',(0.0,0.0,0.0));
#110=DIRECTION('AXIS',(0.0,0.0,1.0));
#111=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#112=COMPOSITE_CURVE('BOUNDARY: PLANAR_FACE3',(#113),.F.);
#113=COMPOSITE_CURVE_SEGMENT(.CONTINUOUS.,.T.,#114);
#114=TRIMMED_CURVE('TRIMMED CURVE FOR CONTOUR OF
PLANAR_FACE3',#115,(#116),(#117),.T.,.CARTESIAN.);
#115=CIRCLE('CIRCLE',#118,21.21);
#116=CARTESIAN_POINT('TRIM POINT 1',(-23.692,21.208,0.0));
#117=CARTESIAN_POINT('TRIM POINT 2',(-23.692,21.208,0.0));
#118=AXIS2_PLACEMENT_3D('CIRCLE PLACEMENT',#119,#120,#121);
#119=CARTESIAN_POINT('CIRCLE CENTER',(-24.0,0.0,0.0));
#120=DIRECTION('Z DIRECTION',(0.0,0.0,1.0));
#121=DIRECTION('X DIRECTION',(1.0,0.0,0.0));
#122=SLOT('SLOT1',#3,(#123),#124,#125,#126,#127,());
#123=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'SLOT1',$,$,#89,#128,#129,$,#
130,#131,#132,$,$,$,$);
#124=AXIS2_PLACEMENT_3D('SLOT1 PLACEMENT',#133,#134,#135);
#125=ELEMENTARY_SURFACE('SLOT1 DEPTH PLANE',#136);
```

```
#126=LINEAR_PATH(#140,#141,#142);
#127=SQUARE_U_PROFILE(#146,#147,#148,0.0,$,0.0);
#128=MILLING_TECHNOLOGY(0.0133,.TCP.,$,83.35,$,$,$,$,$);
#129=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#130=PLUNGE_RAMP($,$);
#131=PLUNGE_RAMP($,$);
#132=BIDIRECTIONAL($,$,$,$,$);
#133=CARTESIAN_POINT('SLOT1',(-15.0,25.0,0.0));
#134=DIRECTION('AXIS',(0.0,0.0,1.0));
#135=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#136=AXIS2_PLACEMENT_3D('SLOT1 DEPTH',#137,#138,#139);
#137=CARTESIAN_POINT('SLOT1 DEPTH',(80.0,0.0,-4.5));
#138=DIRECTION('AXIS',(0.0,0.0,1.0));
#139=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#140=AXIS2_PLACEMENT_3D('PLACEMENT OF TRAVEL COURCE:
SLOT1',#143,#144,#145);
#141=TOLERANCED_LENGTH_MEASURE(80.0,$);
#142=DIRECTION('DIRECTION: SLOT1',(-1.0,0.0,0.0));
#143=CARTESIAN_POINT('POINTSLOT1',(80.0,0.0,-3.5));
#144=DIRECTION('AXIS',(0.0,0.0,1.0));
#145=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#146=AXIS2_PLACEMENT_3D('PLACEMENT OF SWEPT SHAPE:
SLOT1',#149,#150,#151);
#147=TOLERANCED_LENGTH_MEASURE(6.0,$);
#148=TOLERANCED_LENGTH_MEASURE(0.0,$);
#149=CARTESIAN_POINT('POINTSLOT1',(40.0,0.0,-3.5));
#150=DIRECTION('AXIS',(0.0,0.0,1.0));
#151=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#152=CLOSED_POCKET('POCKET1',#3,(#153),#154,#155,(),$,#156,#157,$,#1
58);
#153=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'POCKET1',$,$,#159,#160,#161,
$,#162,#163,#164,$,$,$,$);
#154=AXIS2_PLACEMENT_3D('POCKET1 PLACEMENT',#167,#168,#169);
#155=ELEMENTARY_SURFACE('POCKET1 DEPTH PLANE',#170);
#156=PLANAR_POCKET_BOTTOM_CONDITION();
#157=TOLERANCED_LENGTH_MEASURE(0.0,$);
#158=CIRCULAR_CLOSED_PROFILE(#174,#175);
#159=MILLING_CUTTING_TOOL('T5',#165,(),$,$,$);
#160=MILLING_TECHNOLOGY(0.0133,.TCP.,$,50.0,$,$,$,$,$);
#161=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#162=PLUNGE_TOOLAXIS($);
#163=PLUNGE_TOOLAXIS($);
#164=CONTOUR_PARALLEL($,$,$,$);
#165=ENDMILL(#166,$,$,$,$);
#166=MILLING_TOOL_DIMENSION(5.0,$,$,$,0.0,$,$);
#167=CARTESIAN_POINT('POCKET1',(18.0,25.0,-7.0));
#168=DIRECTION('AXIS',(0.0,0.0,1.0));
#169=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
```

```
#170=AXIS2_PLACEMENT_3D('POCKET1 DEPTH',#171,#172,#173);
#171=CARTESIAN_POINT('POCKET1 DEPTH',(0.0,0.0,0.0));
#172=DIRECTION('AXIS',(0.0,0.0,1.0));
#173=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#174=AXIS2_PLACEMENT_3D('CIRCULAR PROFILE LOCATION',#176,#177,#178);
#175=TOLERANCED_LENGTH_MEASURE(20.0,$);
#176=CARTESIAN_POINT('LOCATION POINT',(0.0,0.0,0.0));
#177=DIRECTION('AXIS',(0.0,0.0,1.0));
#178=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#179=AXIS2_PLACEMENT_3D('SETUP ORIGIN',#181,#182,#183);
#180=WORKPIECE_SETUP(#3,#184,$,$,());
#181=CARTESIAN_POINT('SETUP LOCATION',(0.0,0.0,0.0));
#182=DIRECTION('AXIS',(0.0,0.0,1.0));
#183=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#184=AXIS2_PLACEMENT_3D('WORKPIECE50.0X50.0X60.0
SETUP',#185,#186,#187);
#185=CARTESIAN_POINT('WORKPIECE50.0X50.0X60.0SETUP LOCATION',(-
25.0,-25.0,0.0));
#186=DIRECTION('AXIS',(0.0,0.0,1.0));
#187=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
ENDSEC;

END-ISO-10303-21;
```

12.4 STEP-NC code for Turn-Mill centre part 1

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('GENERATED FOR ISO 14649-12 EXAMPLE 1',
        'SIMPLE PRORGRAM '),        '1');
FILE_NAME('EXAMPLE1.STP',
        '2013-04-18',
        ('MEHRDAD SAFAIEH'),
        ('UNIVERSITY OF BATH, BATH','UK'),
        $,
        'ISO 14649',
        $);
FILE_SCHEMA(('MACHINING_SCHEMA','MILLING_SCHEMA','TURNING_SCHEMA'));
ENDSEC;

DATA;

// WORKPIECE DEFINITION //

#1=WORKPIECE('SIMPLE WORKPIECE',$,$,$,$,#2,());
#2=RIGHT_CIRCULAR_CYLINDER('WORKPIECE PIECE', #3,50.0,30.0);
#3=AXIS1_PLACEMENT('WORKPIECE PIECE PLACEMENT',#4,#5);
#4=CARTESIAN_POINT('WORKPIECE PIECE: LOCATION
',(0.000,0.000,0.000));
#5=DIRECTION(' AXIS ',(0.000,0.000,1.000));

// MANUFACTURING FEATURES //

#10=REVOLVED_FLAT('REVOLVED FLAT',#1,(#20,#21),#100,$,0.0,#104);
#11=GENERAL_REVOLUTION('GENERAL_REVOLUTION',#1,(#22,#23),#126,$,30.0
,#130);
#12=GENERAL_OUTSIDE_PROFILE('NGON PROFILE WITH 4
SIDES',#1,(#24,#25),#109,#113,#118,#123);
#13=CLOSED_POCKET('POCKET',#1,(#24,#25),#215,#219,(),$,#224,$,#225,#
226);
#13=SLOT('SLOT',#1,(#24,#25),#300,#304,#309,#316,());

// OPERATIONS //

#20=FACING_ROUGH($,$,'ROUGH FACE',$,$,#280,#63,#60,$,$,$,0.500);
#21=FACING_FINISH($,$,'FINISH FACE',$,$,#280,#63,#60,$,$,$,0.0);
#22=CONTOURING_ROUGH($,$,'ROUGH GENERAL
REVOLUTION',$,$,#280,#61,#60,$,$,$,0.5);
#23=CONTOURING_FINISH($,$,'FINISH GENERAL
REVOLUTION',$,$,#280,#61,#60,$,$,$,0.0);
#24=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGH
NGON4',$,$,#293,#66,#65,$,$,$,$,$,$,1.000,0.500);
```

```
#25=BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISH
NGON4',$,$,#293,#66,#65,$,$,$,$,$,$,$,$);

// PROJECT  //

#34=PROJECT('TURN-MILL EXAMPLE 1',#35,(#1),$,$,$);
#35=WORKPLAN('MAIN
WORKPLAN',(#36,#37,#38,#39,#40,#41,#42,#43,#44,#45),$,#52,$);
#36=MACHINING_WORKINGSTEP('ROUGH FACE',#54,#10,#20);
#37=MACHINING_WORKINGSTEP('FINISH FACE',#54,#10,#21);
#38=MACHINING_WORKINGSTEP('ROUGH CONTOURING GENERAL
REVOLUTION',#54,#11,#22);
#39=MACHINING_WORKINGSTEP('FINISH CONTOURING GENERAL
REVOLUTION',#54,#11,#23);
#40=MACHINING_WORKINGSTEP('ROUGH BOTTOM AND SIDE MILLING NGON 4
SIDE',#54,#12,#24);
#41=MACHINING_WORKINGSTEP('FINISH BOTTOM AND SIDE MILLING NGON 4
SIDE',#54,#12,#25);
#42=MACHINING_WORKINGSTEP('ROUGH POCKET',#54,#13,#24);
#43=MACHINING_WORKINGSTEP('FINISH POCKET',#54,#13,#25);
#44=MACHINING_WORKINGSTEP('ROUGH SLOT',#54,#13,#24);
#45=MACHINING_WORKINGSTEP('FINISH SLOT',#54,#13,#25);

#52=SETUP('SETUP',$,#54,(#53));
#53=WORKPIECE_SETUP(#1,#70,$,$,$);
#54=PLANE('SECURITY PLANE',#55);
#55=AXIS2_PLACEMENT_3D('SECURITY PLANE',#56,$,$);
#56=CARTESIAN_POINT('SECPLANE: LOCATION',(120.000,0.000,200.000));
#70=AXIS2_PLACEMENT_3D('WORKPIECE',#71,#72,#73);
#71=CARTESIAN_POINT('WORKPIECE: LOCATION',(0.000,0.000,0.000));
#72=DIRECTION('WORKPIECE: AXIS',(0.000,0.000,1.000));
#73=DIRECTION('WORKPIECE: REF_DIRECTION',(1.000,0.000,0.000));

// FUNCTIONS AND TECHNOLOGY //

#60=TURNING_MACHINE_FUNCTIONS(.T.,$,$,(),.F.,$,$,(),$,$,$);
#61=TURNING_TECHNOLOGY($,.TCP.,#62,0.300,.F.,.F.,.F.,$);
#62=CONST_SPINDLE_SPEED(500);
#63=TURNING_TECHNOLOGY($,.TCP.,#64,0.300,.F.,.F.,.F.,$);
#64=CONST_SPINDLE_SPEED(500);
#65= MILLING_MACHINE_FUNCTIONS(.T.,$,$,.F.,$,(),.T.,$,$,());
#66= MILLING_TECHNOLOGY(0.030,.TCP.,#67,16.000,$,.F.,.F.,.F.,$);
#67=CONST_SPINDLE_SPEED(200);

// PLACEMENTS AND LENGTHS //

#100=AXIS2_PLACEMENT_3D('PLACEMENT REVOLVED FLAT',#101,#102,#103);
```

```
#101=CARTESIAN_POINT('REVOLVED FLAT: LOCATION
',(0.000,0.000,0.000));
#102=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#103=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#104=LINEAR_PROFILE('REVOLVED_FLAT_RADIUS',#105,30.000);
#105=AXIS2_PLACEMENT_3D('PLACEMENT',#106,#107,#108);
#106=CARTESIAN_POINT('LOCATION ',(0.000,0.000,0.000));
#107=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#108=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));

#109=AXIS2_PLACEMENT_3D('PLACEMENT NGON PROFILE WITH 4
SIDES',#110,#111,#112);
#110=CARTESIAN_POINT('NGON PROFILE WITH 4 SIDES: LOCATION
',(25.000,0.000,0.000));
#111=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#112=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#113=ELEMENTARY_SURFACE('DEPTH SURFACE FOR NGON PROFILE WITH 4
SIDES',#114);
#114=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 4
SIDES',#115,#116,#117);
#115=CARTESIAN_POINT('NGON PROFILE WITH 4 SIDES: DEPTH
',(0.000,0.000,-15.000));
#116=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#117=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#118=LINEAR_PATH('NGON PROFILE WITH 4 SIDES',#119);
#119=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 4
SIDES',#120,#121,#122);
#120=CARTESIAN_POINT('NGON PROFILE WITH 4 SIDES SWEPT SHAPE:
DISTANCE',(36.44));
#121=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#122=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#123=NGON_PROFILE($,#124,4,.F.);
#124=TOLERANCED_LENGTH_MEASURE(25.000,#125);
#125=PLUS_MINUS_VALUE(0.100,0.100,3);

#126=AXIS2_PLACEMENT_3D('PLACEMENT
GENERAL_REVOLUTION',#127,#128,#129);
#127=CARTESIAN_POINT('GENERAL_REVOLUTION: LOCATION ',(30.000,0.000,-
15.000));
#128=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#129=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#130=GENERAL_PROFILE($,#131);
#131=POLYLINE('',(#132,#133));
#132=CARTESIAN_POINT('',(30.000,0.000, 15.000));
#133=CARTESIAN_POINT('',(30.000,0.000, 35.000));

#215=AXIS2_PLACEMENT_3D('POCKET',#216,#217,#218);
#216=CARTESIAN_POINT('POCKET:LOCATION ',(12.500,0.000,0.000));
```

```
#217=DIRECTION('AXIS ',(0.000,0.000,1.000));
#218=DIRECTION('REF_DIRECTION',(-1.000,0.000,0.000));
#219=ELEMENTARY_SURFACE('DEPTH SURFACE FOR POCKET',#220);
#220=AXIS2_PLACEMENT_3D('POCKET',#221,#222,#223);
#221=CARTESIAN_POINT('POCKET1:DEPTH ',(0.000,0.000,10.000));
#222=DIRECTION('AXIS ',(0.000,0.000,1.000));
#223=DIRECTION('REF_DIRECTION',(1.000,0.000,0.000));
#224=PLANAR_POCKET_BOTTOM_CONDITION();
#225=TOLERANCED_LENGTH_MEASURE(2.50,#125);
#226=CIRCULAR_CLOSED_PROFILE($,#227);
#227=TOLERANCED_LENGTH_MEASURE(25.000,#125);

#300=AXIS2_PLACEMENT_3D('SLOT PLACEMENT',#301,#302,#303);
#301=CARTESIAN_POINT('SLOT1',(-18.220,5.000,0.0));
#302=DIRECTION('AXIS',(0.0,0.0,1.0));
#303=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#304=ELEMENTARY_SURFACE('SLOT DEPTH PLANE',#305);
#305=AXIS2_PLACEMENT_3D('SLOT DEPTH',#306,#307,#308);
#306=CARTESIAN_POINT('SLOT DEPTH',(0.000,0.000,-5.000));
#307=DIRECTION('AXIS',(0.0,0.0,1.0));
#308=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#309=LINEAR_PATH(#310,#311,#312);
#310=AXIS2_PLACEMENT_3D('PLACEMENT OF TRAVEL COURCE:
SLOT',#313,#314,#315);
#311=TOLERANCED_LENGTH_MEASURE(36.44.0,$);
#312=DIRECTION('DIRECTION: SLOT1',(-1.0,0.0,0.0));
#313=CARTESIAN_POINT('POINTSLOT1',(36.44,0.0,-5));
#314=DIRECTION('AXIS',(0.0,0.0,1.0));
#315=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#316=SQUARE_U_PROFILE(#317,#318,#319,0.0,$,0.0);
#317=AXIS2_PLACEMENT_3D('PLACEMENT OF SWEPT SHAPE:
SLOT',#320,#321,#322);
#318=TOLERANCED_LENGTH_MEASURE(10.000,$);
#319=TOLERANCED_LENGTH_MEASURE(0.000,$);
#320=CARTESIAN_POINT('POINTSLOT1',(40.0,0.0,-5));
#321=DIRECTION('AXIS',(0.0,0.0,1.0));
#322=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
// TOOLS //
#280=TURNING_MACHINE_TOOL('',#281,(#283),120,40,$);
#281=GENERAL_TURNING_TOOL(#282,.LEFT.,40,60,.CW.);
#282=TOOL_DIMENSION($,$,$,$,25,5,7,3,5,0.5,$);
#283=CUTTING_COMPONENT(0.000000,$,$,$,$);
#293= MILLING_CUTTING_TOOL('ENDMILL 6MM',#294,(#296),100.000,$,$);
#294= ENDMILL(#295,6,.RIGHT.,.F.,$);
#295= MILLING_TOOL_DIMENSION(6.000,$,$,$,$,$,$);
#296= CUTTING_COMPONENT(100.000,$,$,$,$);
ENDSEC;
END-ISO-10303-21;
```

12.5 STEP-NC code for Milling centre part 2

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('GENERATED ISO 14649-11 FILE','EXAMPLE OF NC
PROGRAMME FOR MILLING: EXAMPLE 2'), '1');
FILE_NAME('EXAMPLE 2.STP', '2013-04-17', ('MEHRDAD
SAFAIEH'),('UNIVERSITY OF BATH, BATH,UK'),$,'ISO 14649',$);
FILE_SCHEMA(('MACHINING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;

DATA;
#1=PROJECT('MILLING EXAMPLE 2',#2,(#3),$,$,$);
#2=WORKPLAN('MAIN
WORKPLAN',(#4,#5,#6,#7,#8,#9,#10,#11,#12,#13,#14,#15,#16),$,#17,$);
#3=WORKPIECE('WORKPIECE',$,$,$,$,#23,());

#4=MACHINING_WORKINGSTEP('PLANAR_FACE',#18,#28,#38,$);
#5=MACHINING_WORKINGSTEP('ROUGHING PROFILE_FEATURE HGON
3',#18,#37,#41,$);
#6=MACHINING_WORKINGSTEP('FINISHING PROFILE_FEATURE HGON
3',#18,#37,#42,$);
#7=MACHINING_WORKINGSTEP('ROUGHING PROFILE_FEATURE CIRCULAR
CLOSED',#18,#30,#39,$);
#8=MACHINING_WORKINGSTEP('FINISHING PROFILE_FEATURE CIRCULAR
CLOSED',#18,#30,#40,$);
#9=MACHINING_WORKINGSTEP('ROUGHING PROFILE_FEATURE HGON
6',#18,#29,#39,$);
#10=MACHINING_WORKINGSTEP('FINISHING PROFILE_FEATURE HGON
6',#18,#29,#40,$);
#11=MACHINING_WORKINGSTEP('DRILLING HOLE 1',#18,#31,#43,$);
#12=MACHINING_WORKINGSTEP('DRILLING HOLE 2',#18,#32,#44,$);
#13=MACHINING_WORKINGSTEP('DRILLING HOLE 3',#18,#33,#43,$);
#14=MACHINING_WORKINGSTEP('DRILLING HOLE 4',#18,#34,#44,$);
#15=MACHINING_WORKINGSTEP('DRILLING HOLE 5',#18,#35,#43,$);
#16=MACHINING_WORKINGSTEP('DRILLING HOLE 6',#18,#36,#44,$);
#17=SETUP('SETUP',#50,#18,(#54));
#18=ELEMENTARY_SURFACE('SECURITY PLANE',#19);
#19=AXIS2_PLACEMENT_3D('SECURITY PLANE PLACEMENT',#20,#21,#22);
#20=CARTESIAN_POINT('SECURITY PLANE: LOCATION',(-25.0,-25.0,-
15.0,0.0,0.0));
#21=DIRECTION('AXIS',(0.0,0.0,1.0));
#22=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));

#23=RIGHT_CIRCULAR_CYLINDER('WORKPIECE PIECE',#24,65.0, 25.0);
#24=AXIS1_PLACEMENT('WORKPIECE PIECE PLACEMENT',#25,#26,#27);
#25=CARTESIAN_POINT('WORKPIECE PIECE: LOCATION
',(0.000,0.000,0.000));
#26=DIRECTION('AXIS',(0.000,0.000,1.000));
```

```
#27=DIRECTION('REF_DIRECTION',(1.000,0.000,0.000));

#28=PLANAR_FACE('PLANAR_FACE',#3,(#38),#100,#104,#220,#224,$,());
#29=GENERAL_OUTSIDE_PROFILE('NGON PROFILE WITH 6
SIDES',#3,(#39,#40),#109,#113,#118,#123);
#30=GENERAL_OUTSIDE_PROFILE('CIRCULAR_CLOSED_PROFILE',#3,(#39,#40),#
126,#230,#235,#240);
#31=ROUND_HOLE('HOLE1 FLAT BOTTOM ',#3,(#43),#134,#138,#143,$,#144);
#32=ROUND_HOLE('HOLE2 CONICAL BOTTOM
',#3,(#44),#145,#149,#154,$,#155);
#33=ROUND_HOLE('HOLE3 FLAT BOTTOM ',#3,(#43),#156,#160,#165,$,#166);
#34=ROUND_HOLE('HOLE4 CONICAL BOTTOM
',#3,(#44),#167,#171,#176,$,#177);
#35=ROUND_HOLE('HOLE5 FLAT BOTTOM ',#3,(#43),#178,#182,#187,$,#188);
#36=ROUND_HOLE('HOLE6 CONICAL BOTTOM
',#3,(#44),#189,#193,#198,$,#199);
#37=GENERAL_OUTSIDE_PROFILE('NGON PROFILE WITH 3
SIDES',#3,(#41,#42),#200,#204,#209,#214);

#38=PLANE_FINISH_MILLING($,$,'PLANAR_FACE',$,$,#300,#301,#302,$,#303
,#304,#305,$,$);
#39=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'PROFILE_FEATURE WITH
12MM',$,$,#310,#311,#312,$,#313,#314,#315,$,$,0.5,0.5);
#40=BOTTOM_AND_SIDE_FINIDH_MILLING($,$,'PROFILE_FEATURE WITH
12MM',$,$,#310,#311,#312,$,#313,#314,#315,$,$,0.000,0.000);
#41=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'PROFILE_FEATURE WITH
6MM',$,$,#320,#321,#322,$,#323,#324,#325,$,$,0.5,0.5);
#42=BOTTOM_AND_SIDE_FINIDH_MILLING($,$,'PROFILE_FEATURE WITH
6MM',$,$,#320,#321,#322,$,#323,#324,#325,$,$,0.000,0.000);
#43=DRILLING($,$,'DRILL HOLE FLAT
END',$,$,#340,#341,#342,$,$,$,$,$,$);
#44=DRILLING($,$,'DRILL HOLE CONICAL
END',$,$,#330,#331,#332,$,$,$,$,$,$);

#50=AXIS2_PLACEMENT_3D('SETUP',#51,#52,#53);
#51=CARTESIAN_POINT('SETUP1: LOCATION ',(0.000,0.000,0.000));
#52=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#53=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#54=WORKPIECE_SETUP(#3,#55,$,$,());
#55=AXIS2_PLACEMENT_3D('WORKPIECE',#56,#57,#58);
#56=CARTESIAN_POINT('WORKPIECE1:LOCATION ',(0.000,0.000,0.000));
#57=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#58=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));

#100=AXIS2_PLACEMENT_3D('PLANAR_FACE',#101,#102,#103);
#101=CARTESIAN_POINT('PLANAR_FACE: LOCATION ',(0.000,0.000,0.000));
#102=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#103=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
```

```
#104=ELEMENTARY_SURFACE('PLANAR_FACE_DEPTH',#105);
#105=AXIS2_PLACEMENT_3D('PLACEMENT',#106,#107,#108);
#106=CARTESIAN_POINT('DEPTH',(0.000,0.000,0.000));
#107=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#108=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#220=LINEAR_PATH($,#221,#222);
#221=TOLERANCED_LENGTH_MEASURE(50.000,#223);
#222=DIRECTION('PLANAR FACE DIRECTION',(0.000,0.000,1.000));
#223=PLUS_MINUS_VALUE(0.300,0.300,3);
#224=LINEAR_PROFILE($,#225);
#225=NUMERIC_PARAMETER('PROFILE LENGTH',50.000,'MM');
#109=AXIS2_PLACEMENT_3D('PLACEMENT NGON PROFILE WITH 6
SIDES',#110,#111,#112);
#110=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES: LOCATION
',(25.000,0.000,-25.000));
#111=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#112=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#113=ELEMENTARY_SURFACE('DEPTH SURFACE FOR NGON PROFILE WITH 6
SIDES',#114);
#114=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 6
SIDES',#115,#116,#117);
#115=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES: DEPTH
',(0.000,0.000,-15.000));
#116=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#117=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#118=LINEAR_PATH('NGON PROFILE WITH 6 SIDES',#119);
#119=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 6
SIDES',#120,#121,#122);
#120=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES SWEPT SHAPE:
DISTANCE',(25.000));
#121=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#122=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#123=NGON_PROFILE($,#124,6,.F.);
#124=TOLERANCED_LENGTH_MEASURE(25.000,#125);
#125=PLUS_MINUS_VALUE(0.100,0.100,3);
#126=AXIS2_PLACEMENT_3D('PLACEMENT
CIRCULAR_CLOSED_PROFILE',#127,#128,#129);
#127=CARTESIAN_POINT('CIRCULAR_CLOSED_PROFILE: LOCATION
',(25.000,0.000,-10.000));
#128=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#129=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#230=ELEMENTARY_SURFACE('DEPTH SURFACE FOR
CIRCULAR_CLOSED_PROFILE',#231);
#231=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT
CIRCULAR_CLOSED_PROFILE',#232,#233,#234);
#232=CARTESIAN_POINT('CIRCULAR_CLOSED_PROFILE: DEPTH
',(0.000,0.000,-15.000));
#233=DIRECTION(' AXIS ',(1.000,0.000,0.000));
```

```
#234=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#235=LINEAR_PATH('CIRCULAR_CLOSED_PROFILE',#236);
#236=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT
CIRCULAR_CLOSED_PROFILE',#237,#238,#239);
#237=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES SWEPT SHAPE:
DISTANCE',(0.000,0.000,-15.000));
#238=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#239=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#240=CIRCULAR_CLOSED_PROFILE($,#241);
#241=TOLERANCED_LENGTH_MEASURE(50.000,#125);

#134=AXIS2_PLACEMENT_3D('HOLE1',#135,#136,#137);
#135=CARTESIAN_POINT('HOLE1: LOCATION ',(18.75,10.82,-32.5));
#136=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#137=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#138=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE1',#139);
#139=AXIS2_PLACEMENT_3D('HOLE1 DEPTH',#140,#141,#142);
#140=CARTESIAN_POINT('HOLE1: DEPTH ',(0.000,0.000,-5.000));
#141=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#142=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#143=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#144=FLAT_HOLE_BOTTOM();
#145=AXIS2_PLACEMENT_3D('HOLE2',#146,#147,#148);
#146=CARTESIAN_POINT('HOLE2: LOCATION ',(0.000,21.65,-32.5));
#147=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#148=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#149=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE2',#150);
#150=AXIS2_PLACEMENT_3D('HOLE2 DEPTH',#151,#152,#153);
#151=CARTESIAN_POINT('HOLE2: DEPTH ',(0.000,0.000,-5.000));
#152=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#153=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#154=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#155=CONICAL_HOLE_BOTTOM(30,$);
#156=AXIS2_PLACEMENT_3D('HOLE3',#157,#158,#159);
#157=CARTESIAN_POINT('HOLE1: LOCATION ',(-18.75,10.82,-32.5));
#158=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#159=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#160=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE3',#161);
#161=AXIS2_PLACEMENT_3D('HOLE3 DEPTH',#162,#163,#164);
#162=CARTESIAN_POINT('HOLE3: DEPTH ',(0.000,0.000,-5.000));
#163=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#164=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#165=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#166=FLAT_HOLE_BOTTOM();
#167=AXIS2_PLACEMENT_3D('HOLE4',#168,#169,#170);
#168=CARTESIAN_POINT('HOLE4: LOCATION ',(-18.75,-10.82,-32.5));
#169=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#170=DIRECTION(' AXIS ',(1.000,0.000,0.000));
```

```
#171=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE4',#172);
#172=AXIS2_PLACEMENT_3D('HOLE4 DEPTH',#173,#174,#175);
#173=CARTESIAN_POINT('HOLE4: DEPTH ',(0.000,0.000,-5.000));
#174=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#175=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#176=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#177=CONICAL_HOLE_BOTTOM(30,$);

#178=AXIS2_PLACEMENT_3D('HOLE5',#179,#180,#181);
#179=CARTESIAN_POINT('HOLE5: LOCATION ',(0,-21.65,-32.5));
#180=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#181=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#182=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE5',#183);
#183=AXIS2_PLACEMENT_3D('HOLE5 DEPTH',#184,#185,#186);
#184=CARTESIAN_POINT('HOLE3: DEPTH ',(0.000,0.000,-5.000));
#185=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#186=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#187=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#188=FLAT_HOLE_BOTTOM();
#189=AXIS2_PLACEMENT_3D('HOLE6',#190,#191,#192);
#190=CARTESIAN_POINT('HOLE6: LOCATION ',(18.75,-10.82,-32.5));
#191=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#192=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#193=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE6',#194);
#194=AXIS2_PLACEMENT_3D('HOLE6 DEPTH',#195,#196,#197);
#195=CARTESIAN_POINT('HOLE6: DEPTH ',(0.000,0.000,-5.000));
#196=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#197=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#198=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#199=CONICAL_HOLE_BOTTOM(30,$);
#200=AXIS2_PLACEMENT_3D('PLACEMENT NGON PROFILE WITH 3
SIDES',#201,#202,#203);
#201=CARTESIAN_POINT('NGON PROFILE WITH 3 SIDES: LOCATION
',(25.000,0.000,0.000));
#202=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#203=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#204=ELEMENTARY_SURFACE('DEPTH SURFACE FOR NGON PROFILE WITH 3
SIDES',#205);
#205=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 3
SIDES',#206,#207,#208);
#206=CARTESIAN_POINT('NGON PROFILE WITH 3 SIDES: DEPTH
',(0.000,0.000,-10.000));
#207=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#208=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#209=LINEAR_PATH('NGON PROFILE WITH 3 SIDES',#210);
#210=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 3
SIDES',#211,#212,#213);
```

```
#211=CARTESIAN_POINT('NGON PROFILE WITH 3 SIDES SWEPT SHAPE:
DISTANCE',(43.3));
#212=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#213=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#214=NGON_PROFILE($,#215,3,.F.);
#215=TOLERANCED_LENGTH_MEASURE(25.000,#125);
#300=MILLING_CUTTING_TOOL('FACE MILL 60MM',#306,(),$,$,$);
#301=MILLING_TECHNOLOGY(0.0166,.TCP.,$,50.0,$,$,$,$,$);
#302=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#303=PLUNGE_RAMP($,$);
#304=PLUNGE_RAMP($,$);
#305=UNIDIRECTIONAL($,$,$,$);
#306=FACEMILL(#307,$,$,$,$);
#307=MILLING_TOOL_DIMENSION(60.0,$,$,$,0.0,$,$);
#310=MILLING_CUTTING_TOOL('SLOT DRILL 12MM',#316,(),$,$,$);
#311=MILLING_TECHNOLOGY(0.0133,.TCP.,$,66.666,$,$,$,$,$);
#312=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#313=PLUNGE_RAMP($,$);
#314=PLUNGE_RAMP($,$);
#315=CONTOUR_PARALLEL($,$,$,$);
#316=ENDMILL(#317,$,$,$,$);
#317=MILLING_TOOL_DIMENSION(12.0,$,$,$,0.0,$,$);
#320=MILLING_CUTTING_TOOL('SLOT DRILL 6MM',#326,(),$,$,$);
#321=MILLING_TECHNOLOGY(0.0133,.TCP.,$,66.666,$,$,$,$,$);
#322=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#323=PLUNGE_RAMP($,$);
#324=PLUNGE_RAMP($,$);
#325=CONTOUR_PARALLEL($,$,$,$);
#326=ENDMILL(#327,$,$,$,$);
#327=MILLING_TOOL_DIMENSION(6.0,$,$,$,0.0,$,$);
#330=MILLING_CUTTING_TOOL('SPIRAL_DRILL_10MM',#333,(),$,$,$);
#331=MILLING_TECHNOLOGY(0.0133,.TCP.,$,66.666,$,$,$,$,$);
#332=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#333=TWIST_DRILL(#334,2,.RIGHT.,.F.,0.840);
#234=MILLING_TOOL_DIMENSION(10.000,45,$,$,$,$,$);

#340=MILLING_CUTTING_TOOL('ENDMILL 10MM',#343,(),$,$,$);
#341=MILLING_TECHNOLOGY(0.0133,.TCP.,$,66.666,$,$,$,$,$);
#342=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#343=ENDMILL(#344,6,.RIGHT.,.F.,$);
#344=MILLING_TOOL_DIMENSION(10.000,$,$,$,$,$,$);
ENDSEC;
END-ISO-10303-21;
```

12.6 STEP-NC code for Turn-Mill centre part 2

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('EXAMPLE OF NC PROGRAMME FOR TURN-MILLING: EXAMPLE
2.'),'1');
FILE_NAME('EXAMPLE2.STP',$,('ISO14649'),(''),'MEHRDAD
SAFAIEH','UNIVERSITY OF BATH', 'BATH');
FILE_SCHEMA(('MACHINING_SCHEMA','TURNING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;
DATA;

// WORKPIECE DEFINITION //

#1=WORKPIECE('SIMPLE WORKPIECE',$,$,$,$,#2,());
#2=RIGHT_CIRCULAR_CYLINDER('WORKPIECE PIECE', #3,65.0, 25.0);
#3=AXIS1_PLACEMENT('WORKPIECE PIECE PLACEMENT',#4,#5);
#4=CARTESIAN_POINT('WORKPIECE PIECE: LOCATION
',(0.000,0.000,0.000));
#5=DIRECTION(' AXIS ',(0.000,0.000,1.000));

// MANUFACTURING FEATURES //

#10=REVOLVED_FLAT('REVOLVED FLAT',#1,(#20,#21),#100,$,0.0,#104);
#11=GENERAL_OUTSIDE_PROFILE('NGON PROFILE WITH 6
SIDES',#1,(#26,#27),#109,#113,#118,#123);
#12=GENERAL_REVOLUTION('GENERAL_REVOLUTION',#1,(#22,#23),#126,$,25.0
,#130);
#13=ROUND_HOLE('HOLE1 FLAT BOTTOM ',#1,(#24),#134,#138,#143,$,#144);
#14=ROUND_HOLE('HOLE2 CONICAL BOTTOM
',#1,(#25),#145,#149,#154,$,#155);
#15=ROUND_HOLE('HOLE3 FLAT BOTTOM ',#1,(#24),#156,#160,#165,$,#166);
#16=ROUND_HOLE('HOLE4 CONICAL BOTTOM
',#1,(#25),#167,#171,#176,$,#177);
#17=ROUND_HOLE('HOLE5 FLAT BOTTOM ',#1,(#24),#178,#182,#187,$,#188);
#18=ROUND_HOLE('HOLE6 CONICAL BOTTOM
',#1,(#25),#189,#193,#198,$,#199);
#19=GENERAL_OUTSIDE_PROFILE('NGON PROFILE WITH 3
SIDES',#1,(#28,#29),#200,#204,#209,#214);

// OPERATIONS //

#20=FACING_ROUGH($,$,'ROUGH FACE',$,$,#280,#63,#60,$,$,$,0.500);
#21=FACING_FINISH($,$,'FINISH FACE',$,$,#280,#63,#60,$,$,$,0.0);
#22=CONTOURING_ROUGH($,$,'ROUGH GENERAL
REVOLUTION',$,$,#280,#61,#60,$,$,$,0.5);
#23=CONTOURING_FINISH($,$,'FINISH GENERAL
REVOLUTION',$,$,#280,#61,#60,$,$,$,0.0);
```

```
#24=DRILLING($,$,'DRILL HOLE FLAT
END',$,$,#293,#66,#65,$,$,$,$,$,$);
#25=DRILLING($,$,'DRILL HOLE CONICAL
END',$,$,#289,#66,#65,$,$,$,$,$,$);
#26=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGH
NGON6',$,$,#293,#66,#65,$,$,$,$,$,$,1.000,0.500);
#27=BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISH
NGON6',$,$,#293,#66,#65,$,$,$,$,$,$,$,$);
#28=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGH
NGON3',$,$,#293,#66,#65,$,$,$,$,$,$,1.000,0.500);
#29=BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISH
NGON3',$,$,#293,#66,#65,$,$,$,$,$,$,$,$);

// PROJECT  //

#34=PROJECT('TURN-MILL EXAMPLE 2',#35,(#1),$,$,$);
#35=WORKPLAN('MAIN
WORKPLAN',(#38,#39,#40,#41,#42,#43,#44,#45,#46,#47,#49,#50,#51),$,#5
2,$);
#38=MACHINING_WORKINGSTEP('ROUGH FACE',#54,#10,#20,$);
#39=MACHINING_WORKINGSTEP('FINISH FACE',#54,#10,#21,$);
#40=MACHINING_WORKINGSTEP('ROUGH CONTOURING GENERAL
REVOLUTION',#54,#12,#22,$);
#41=MACHINING_WORKINGSTEP('FINISH CONTOURING GENERAL
REVOLUTION',#54,#12,#23,$);
#42=MACHINING_WORKINGSTEP('ROUGH BOTTOM AND SIDE MILLING NGON 6
SIDE',#54,#11,#26,$);
#43=MACHINING_WORKINGSTEP('FINISH BOTTOM AND SIDE MILLING NGON 6
SIDE',#54,#11,#27,$);
#44=MACHINING_WORKINGSTEP('DRILLING HOLE 1',#54,#13,#24,$);
#45=MACHINING_WORKINGSTEP('DRILLING HOLE 2',#54,#14,#25,$);
#46=MACHINING_WORKINGSTEP('DRILLING HOLE 3',#54,#15,#24,$);
#47=MACHINING_WORKINGSTEP('DRILLING HOLE 4',#54,#16,#25,$);
#48=MACHINING_WORKINGSTEP('DRILLING HOLE 5',#54,#17,#24,$);
#49=MACHINING_WORKINGSTEP('DRILLING HOLE 6',#54,#18,#25,$);
#50=MACHINING_WORKINGSTEP('ROUGH BOTTOM AND SIDE MILLING NGON 3
SIDE',#54,#19,#28,$);
#51=MACHINING_WORKINGSTEP('FINISH BOTTOM AND SIDE MILLING NGON 3
SIDE',#54,#19,#29,$);
#52=SETUP('SETUP',$,#54,(#53));
#53=WORKPIECE_SETUP(#1,#70,$,$,$);
#54=PLANE('SECURITY PLANE',#55);
#55=AXIS2_PLACEMENT_3D('SECURITY PLANE',#56,$,$);
#56=CARTESIAN_POINT('SECPLANE: LOCATION',(120.000,0.000,200.000));
#70=AXIS2_PLACEMENT_3D('WORKPIECE',#71,#72,#73);
#71=CARTESIAN_POINT('WORKPIECE: LOCATION',(0.000,0.000,0.000));
#72=DIRECTION('WORKPIECE: AXIS',(0.000,0.000,1.000));
#73=DIRECTION('WORKPIECE: REF_DIRECTION',(1.000,0.000,0.000));
```

```
// FUNCTIONS AND TECHNOLOGY //

#60=TURNING_MACHINE_FUNCTIONS(.T.,$,$,(),.F.,$,$,(),$,$,$);
#61=TURNING_TECHNOLOGY($,.TCP.,#62,0.300,.F.,.F.,.F.,$);
#62=CONST_SPINDLE_SPEED(500);
#63=TURNING_TECHNOLOGY($,.TCP.,#64,0.300,.F.,.F.,.F.,$);
#64=CONST_SPINDLE_SPEED(500);
#65= MILLING_MACHINE_FUNCTIONS(.T.,$,$,.F.,$,(),.T.,$,$,());
#66= MILLING_TECHNOLOGY(0.030,.TCP.,#67,16.000,$,.F.,.F.,.F.,$);
#67=CONST_SPINDLE_SPEED(200);

// PLACEMENTS AND LENGTHS //

#100=AXIS2_PLACEMENT_3D('PLACEMENT REVOLVED FLAT',#101,#102,#103);
#101=CARTESIAN_POINT('REVOLVED FLAT: LOCATION
',(0.000,0.000,0.000));
#102=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#103=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#104=LINEAR_PROFILE('REVOLVED_FLAT_RADIUS',#105,25.000);
#105=AXIS2_PLACEMENT_3D('PLACEMENT',#106,#107,#108);
#106=CARTESIAN_POINT('LOCATION ',(0.000,0.000,0.000));
#107=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#108=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));

#109=AXIS2_PLACEMENT_3D('PLACEMENT NGON PROFILE WITH 6
SIDES',#110,#111,#112);
#110=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES: LOCATION
',(25.000,0.000,-25.000));
#111=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#112=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#113=ELEMENTARY_SURFACE('DEPTH SURFACE FOR NGON PROFILE WITH 6
SIDES',#114);
#114=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 6
SIDES',#115,#116,#117);
#115=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES: DEPTH
',(0.000,0.000,-15.000));
#116=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#117=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#118=LINEAR_PATH('NGON PROFILE WITH 6 SIDES',#119);
#119=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 6
SIDES',#120,#121,#122);
#120=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES SWEPT SHAPE:
DISTANCE',(25.000));
#121=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#122=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#123=NGON_PROFILE($,#124,6,.F.);
#124=TOLERANCED_LENGTH_MEASURE(25.000,#125);
```

```
#125=PLUS_MINUS_VALUE(0.100,0.100,3);

#126=AXIS2_PLACEMENT_3D('PLACEMENT
GENERAL_REVOLUTION',#127,#128,#129);
#127=CARTESIAN_POINT('GENERAL_REVOLUTION: LOCATION ',(25.000,0.000,-
10.000));
#128=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#129=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#130=GENERAL_PROFILE($,#131);
#131=POLYLINE('',(#132,#133));
#132=CARTESIAN_POINT('',(25.000,0.000, 10.000));
#133=CARTESIAN_POINT('',(25.000,0.000, 25.000));

#134=AXIS2_PLACEMENT_3D('HOLE1',#135,#136,#137);
#135=CARTESIAN_POINT('HOLE1: LOCATION ',(18.75,10.82,-32.5));
#136=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#137=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#138=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE1',#139);
#139=AXIS2_PLACEMENT_3D('HOLE1 DEPTH',#140,#141,#142);
#140=CARTESIAN_POINT('HOLE1: DEPTH ',(0.000,0.000,-5.000));
#141=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#142=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#143=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#144=FLAT_HOLE_BOTTOM();

#145=AXIS2_PLACEMENT_3D('HOLE2',#146,#147,#148);
#146=CARTESIAN_POINT('HOLE2: LOCATION ',(0.000,21.65,-32.5));
#147=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#148=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#149=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE2',#150);
#150=AXIS2_PLACEMENT_3D('HOLE2 DEPTH',#151,#152,#153);
#151=CARTESIAN_POINT('HOLE2: DEPTH ',(0.000,0.000,-5.000));
#152=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#153=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#154=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#155=CONICAL_HOLE_BOTTOM(30,$);

#156=AXIS2_PLACEMENT_3D('HOLE3',#157,#158,#159);
#157=CARTESIAN_POINT('HOLE1: LOCATION ',(-18.75,10.82,-32.5));
#158=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#159=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#160=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE3',#161);
#161=AXIS2_PLACEMENT_3D('HOLE3 DEPTH',#162,#163,#164);
#162=CARTESIAN_POINT('HOLE3: DEPTH ',(0.000,0.000,-5.000));
#163=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#164=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#165=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#166=FLAT_HOLE_BOTTOM();
```

```
#167=AXIS2_PLACEMENT_3D('HOLE4',#168,#169,#170);
#168=CARTESIAN_POINT('HOLE4: LOCATION ',(-18.75,-10.82,-32.5));
#169=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#170=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#171=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE4',#172);
#172=AXIS2_PLACEMENT_3D('HOLE4 DEPTH',#173,#174,#175);
#173=CARTESIAN_POINT('HOLE4: DEPTH ',(0.000,0.000,-5.000));
#174=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#175=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#176=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#177=CONICAL_HOLE_BOTTOM(30,$);

#178=AXIS2_PLACEMENT_3D('HOLE5',#179,#180,#181);
#179=CARTESIAN_POINT('HOLE5: LOCATION ',(0,-21.65,-32.5));
#180=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#181=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#182=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE5',#183);
#183=AXIS2_PLACEMENT_3D('HOLE5 DEPTH',#184,#185,#186);
#184=CARTESIAN_POINT('HOLE3: DEPTH ',(0.000,0.000,-5.000));
#185=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#186=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#187=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#188=FLAT_HOLE_BOTTOM();

#189=AXIS2_PLACEMENT_3D('HOLE6',#190,#191,#192);
#190=CARTESIAN_POINT('HOLE6: LOCATION ',(18.75,-10.82,-32.5));
#191=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#192=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#193=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE6',#194);
#194=AXIS2_PLACEMENT_3D('HOLE6 DEPTH',#195,#196,#197);
#195=CARTESIAN_POINT('HOLE6: DEPTH ',(0.000,0.000,-5.000));
#196=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#197=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#198=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#199=CONICAL_HOLE_BOTTOM(30,$);

#200=AXIS2_PLACEMENT_3D('PLACEMENT NGON PROFILE WITH 3
SIDES',#201,#202,#203);
#201=CARTESIAN_POINT('NGON PROFILE WITH 3 SIDES: LOCATION
',(25.000,0.000,0.000));
#202=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#203=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#204=ELEMENTARY_SURFACE('DEPTH SURFACE FOR NGON PROFILE WITH 3
SIDES',#205);
#205=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 3
SIDES',#206,#207,#208);
```

```
#206=CARTESIAN_POINT('NGON PROFILE WITH 3 SIDES: DEPTH
',(0.000,0.000,-10.000));
#207=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#208=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#209=LINEAR_PATH('NGON PROFILE WITH 3 SIDES',#210);
#210=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 3
SIDES',#211,#212,#213);
#211=CARTESIAN_POINT('NGON PROFILE WITH 3 SIDES SWEPT SHAPE:
DISTANCE',(43.300));
#212=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#213=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#214=NGON_PROFILE($,#215,3,.F.);
#215=TOLERANCED_LENGTH_MEASURE(25.000,#125);

// TOOLS //

#280=TURNING_MACHINE_TOOL('',#281,(#283),120,40,$);
#281=GENERAL_TURNING_TOOL(#282,.LEFT.,40,60,.CW.);
#282=TOOL_DIMENSION($,$,$,$,25,5,7,3,5,0.5,$);
#283=CUTTING_COMPONENT(0.000000,$,$,$,$);

#289=
MILLING_CUTTING_TOOL('SPIRAL_DRILL_10MM',#290,(#292),90.000,$,$);
#290= TWIST_DRILL(#291,2,.RIGHT.,.F.,0.840);
#291= MILLING_TOOL_DIMENSION(10.000,45.000,$,$,$,$,$);
#292= CUTTING_COMPONENT(90.000,$,$,$,$);

#293= MILLING_CUTTING_TOOL('ENDMILL 10MM',#294,(#296),100.000,$,$);
#294= ENDMILL(#295,6,.RIGHT.,.F.,$);
#295= MILLING_TOOL_DIMENSION(10.000,$,$,$,$,$,$);
#296= CUTTING_COMPONENT(100.000,$,$,$,$);

ENDSEC;
END-ISO-10303-21;
```

12.7 STEP-NC code for Milling centre part 3

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('GENERATED ISO 14649-11 FILE','EXAMPLE OF NC
PROGRAMME FOR MILLING: EXAMPLE 3'), '1');
FILE_NAME('EXAMPLE 3.STP', '2013-06-17', ('MEHRDAD
SAFAIEH'),('UNIVERSITY OF BATH, BATH,UK'),$,'ISO 14649',$);
FILE_SCHEMA(('MACHINING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;

DATA;
#1=PROJECT('MILLING EXAMPLE 3',#2,(#5),$,$,$);
#2=WORKPLAN('MAIN WORKPLAN',(#3,#4),$,$,$);
#3=WORKPLAN('WORKPLAN
1',(#6,#7,#8,#9,#10,#11,#12,#13,#14,#15),$,#22,$);
#4=WORKPLAN('WORKPLAN 2',(#16,#17,#18,#19,#20,#21),$,#37,$);
#5=WORKPIECE('WORKPIECE',$,$,$,$,#52,());

#6=MACHINING_WORKINGSTEP('PLANAR_FACE',#27,#60,#80,$);
#7=MACHINING_WORKINGSTEP('ROUGHING PROFILE_FEATURE CIRCULAR CLOSED
1',#27,#62,#81,$);
#8=MACHINING_WORKINGSTEP('FINISHING PROFILE_FEATURE CIRCULAR CLOSED
1',#27,#62,#82,$);
#9=MACHINING_WORKINGSTEP('ROUGHING PROFILE_FEATURE HGON
6',#27,#61,#81,$);
#10=MACHINING_WORKINGSTEP('FINISHING PROFILE_FEATURE HGON
6',#27,#61,#82,$);
#11=MACHINING_WORKINGSTEP('DRILLING HOLE 1',#27,#63,#86,$);
#12=MACHINING_WORKINGSTEP('DRILLING HOLE 2',#27,#64,#86,$);
#13=MACHINING_WORKINGSTEP('DRILLING HOLE 3',#27,#65,#86,$);
#14=MACHINING_WORKINGSTEP('DRILLING HOLE 4',#27,#66,#85,$);
#15=MACHINING_WORKINGSTEP('DRILLING HOLE 5',#27,#67,#85,$);
#16=MACHINING_WORKINGSTEP('ROUGH POCKET 1',#27,#69,#83,$);
#17=MACHINING_WORKINGSTEP('FINISH POCKET 1',#27,#69,#84,$);
#18=MACHINING_WORKINGSTEP('ROUGH POCKET 2',#27,#70,#83,$);
#19=MACHINING_WORKINGSTEP('FINISH POCKET 2',#27,#70,#84,$);
#20=MACHINING_WORKINGSTEP('ROUGHING PROFILE_FEATURE CIRCULAR CLOSED
2',#27,#68,#83,$);
#21=MACHINING_WORKINGSTEP('FINISHING PROFILE_FEATURE CIRCULAR CLOSED
2',#27,#68,#84,$);

#22=SETUP('SETUP 1',#23,#27,(#32));
#23=AXIS2_PLACEMENT_3D('SETUP 1',#24,#25,#26);
#24=CARTESIAN_POINT('SETUP1: LOCATION ',(0.000,0.000,0.000));
#25=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#26=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#27=ELEMENTARY_SURFACE('SECURITY PLANE',#28);
```

```
#28=AXIS2_PLACEMENT_3D('SECURITY PLANE PLACEMENT',#29,#30,#31);
#29=CARTESIAN_POINT('SECURITY PLANE: LOCATION',(-25.0,-25.0,-
15.0,0.0,0.0));
#30=DIRECTION('AXIS',(0.0,0.0,1.0));
#31=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#32=WORKPIECE_SETUP(#5,#33,$,$,());
#33=AXIS2_PLACEMENT_3D('WORKPIECE',#34,#35,#36);
#34=CARTESIAN_POINT('WORKPIECE1:LOCATION ',(0.000,0.000,0.000));
#35=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#36=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));

#37=SETUP('SETUP 2',#38,#42,(#47));
#38=AXIS2_PLACEMENT_3D('SETUP 2',#39,#40,#41);
#39=CARTESIAN_POINT('SETUP2: LOCATION ',(0.000,0.000,0.000));
#40=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#41=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#42=ELEMENTARY_SURFACE('SECURITY PLANE',#43);
#43=AXIS2_PLACEMENT_3D('SECURITY PLANE PLACEMENT',#44,#45,#46);
#44=CARTESIAN_POINT('SECURITY PLANE: LOCATION',(-25.0,-25.0,-
15.0,0.0,0.0));
#45=DIRECTION('AXIS',(0.0,0.0,1.0));
#46=DIRECTION('REF_DIRECTION',(1.0,0.0,0.0));
#47=WORKPIECE_SETUP(#5,#48,$,$,());
#48=AXIS2_PLACEMENT_3D('WORKPIECE',#49,#50,#51);
#49=CARTESIAN_POINT('WORKPIECE1:LOCATION ',(0.000,0.000,0.000));
#50=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#51=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));

#52=RIGHT_CIRCULAR_CYLINDER('WORKPIECE PIECE',#53,65.0, 25.0);
#53=AXIS1_PLACEMENT('WORKPIECE PIECE PLACEMENT',#54,#55,#56);
#54=CARTESIAN_POINT('WORKPIECE PIECE: LOCATION
',(0.000,0.000,0.000));
#55=DIRECTION('AXIS',(0.000,0.000,1.000));
#56=DIRECTION('REF_DIRECTION',(1.000,0.000,0.000));

#60=PLANAR_FACE('PLANAR_FACE',#5,(#80),#100,#104,#220,#224,$,());
#61=GENERAL_OUTSIDE_PROFILE('NGON PROFILE WITH 6
SIDES',#5,(#81,#82),#109,#113,#118,#123);
#62=GENERAL_OUTSIDE_PROFILE('CIRCULAR_CLOSED_PROFILE
1',#5,(#81,#82),#126,#230,#235,#240);
#63=ROUND_HOLE('HOLE1 CONICAL
BOTTOM',#5,(#86),#145,#149,#154,$,#155);
#64=ROUND_HOLE('HOLE2 CONICAL
BOTTOM',#5,(#86),#156,#160,#165,$,#166);
#65=ROUND_HOLE('HOLE3 CONICAL
BOTTOM',#5,(#86),#167,#171,#176,$,#177);
#66=ROUND_HOLE('HOLE4 FLAT BOTTOM',#5,(#85),#178,#182,#187,$,#188);
#67=ROUND_HOLE('HOLE5 FLAT BOTTOM',#5,(#85),#189,#193,#198,$,#199);
```

```
#68=GENERAL_OUTSIDE_PROFILE('CIRCULAR_CLOSED_PROFILE
2',#5,(#83,#84),#200,#250,#255,#260);
#69=CLOSED_POCKET('POCKET
1',#5,(#83,#84),#400,#404,(),$,#409,$,#410,#411);
#70=CLOSED_POCKET('POCKET
2',#5,(#83,#84),#417,#421,(),$,#426,$,#427,#428);

#80=PLANE_FINISH_MILLING($,$,'PLANAR_FACE',$,$,#300,#301,#302,$,#303
,#304,#305,$,$);
#81=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'PROFILE_FEATURE WITH
12MM',$,$,#310,#311,#312,$,#313,#314,#315,$,$,0.5,0.5);
#82=BOTTOM_AND_SIDE_FINIDH_MILLING($,$,'PROFILE_FEATURE WITH
12MM',$,$,#310,#311,#312,$,#313,#314,#315,$,$,0.000,0.000);
#83=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'PROFILE_FEATURE WITH
6MM',$,$,#320,#321,#322,$,#323,#324,#325,$,$,0.5,0.5);
#84=BOTTOM_AND_SIDE_FINIDH_MILLING($,$,'PROFILE_FEATURE WITH
6MM',$,$,#320,#321,#322,$,#323,#324,#325,$,$,0.000,0.000);
#85=DRILLING($,$,'DRILL HOLE FLAT
END',$,$,#320,#321,#322,$,$,$,$,$,$);
#86=DRILLING($,$,'DRILL HOLE CONICAL
END',$,$,#330,#331,#332,$,$,$,$,$,$);

#100=AXIS2_PLACEMENT_3D('PLANAR_FACE',#101,#102,#103);
#101=CARTESIAN_POINT('PLANAR_FACE: LOCATION ',(0.000,0.000,0.000));
#102=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#103=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#104=ELEMENTARY_SURFACE('PLANAR_FACE_DEPTH',#105);
#105=AXIS2_PLACEMENT_3D('PLACEMENT',#106,#107,#108);
#106=CARTESIAN_POINT('DEPTH',(0.000,0.000,0.000));
#107=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#108=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#220=LINEAR_PATH($,#221,#222);
#221=TOLERANCED_LENGTH_MEASURE(50.000,#223);
#222=DIRECTION('PLANAR FACE DIRECTION',(0.000,0.000,1.000));
#223=PLUS_MINUS_VALUE(0.300,0.300,3);
#224=LINEAR_PROFILE($,#225);
#225=NUMERIC_PARAMETER('PROFILE LENGTH',50.000,'MM');

#109=AXIS2_PLACEMENT_3D('PLACEMENT NGON PROFILE WITH 6
SIDES',#110,#111,#112);
#110=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES: LOCATION
',(25.000,0.000,0.000));
#111=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#112=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#113=ELEMENTARY_SURFACE('DEPTH SURFACE FOR NGON PROFILE WITH 6
SIDES',#114);
#114=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 6
SIDES',#115,#116,#117);
```

```
#115=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES: DEPTH
',(0.000,0.000,-15.500));
#116=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#117=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#118=LINEAR_PATH('NGON PROFILE WITH 6 SIDES',#119);
#119=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 6
SIDES',#120,#121,#122);
#120=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES SWEPT SHAPE:
DISTANCE',(26.220));
#121=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#122=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#123=NGON_PROFILE($,#124,6,.F.);
#124=TOLERANCED_LENGTH_MEASURE(26.220,#125);
#125=PLUS_MINUS_VALUE(0.100,0.100,3);

#126=AXIS2_PLACEMENT_3D('PLACEMENT
CIRCULAR_CLOSED_PROFILE',#127,#128,#129);
#127=CARTESIAN_POINT('CIRCULAR_CLOSED_PROFILE: LOCATION
',(25.000,0.000,-15.500));
#128=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#129=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#230=ELEMENTARY_SURFACE('DEPTH SURFACE FOR
CIRCULAR_CLOSED_PROFILE',#231);
#231=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT
CIRCULAR_CLOSED_PROFILE',#232,#233,#234);
#232=CARTESIAN_POINT('CIRCULAR_CLOSED_PROFILE: DEPTH
',(0.000,0.000,-12.500));
#233=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#234=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#235=LINEAR_PATH('CIRCULAR_CLOSED_PROFILE',#236);
#236=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT
CIRCULAR_CLOSED_PROFILE',#237,#238,#239);
#237=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES SWEPT SHAPE:
DISTANCE',(0.000,0.000,-12.500));
#238=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#239=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#240=CIRCULAR_CLOSED_PROFILE($,#241);
#241=TOLERANCED_LENGTH_MEASURE(50.000,#125);

#145=AXIS2_PLACEMENT_3D('HOLE1',#146,#147,#148);
#146=CARTESIAN_POINT('HOLE1: LOCATION ',(0.000,0.000,0.000));
#147=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#148=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#149=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE1',#150);
#150=AXIS2_PLACEMENT_3D('HOLE1 DEPTH',#151,#152,#153);
#151=CARTESIAN_POINT('HOLE1: DEPTH ',(0.000,0.000,-13.750));
#152=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#153=DIRECTION(' AXIS ',(1.000,0.000,0.000));
```

```
#154=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#155=CONICAL_HOLE_BOTTOM(49,$);

#156=AXIS2_PLACEMENT_3D('HOLE2',#157,#158,#159);
#157=CARTESIAN_POINT('HOLE2: LOCATION ',(0.000,-17.33,0.000));
#158=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#159=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#160=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE2',#161);
#161=AXIS2_PLACEMENT_3D('HOLE2 DEPTH',#162,#163,#164);
#162=CARTESIAN_POINT('HOLE2: DEPTH ',(0.000,0.000,-13.750));
#163=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#164=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#165=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#166=CONICAL_HOLE_BOTTOM(49,$);

#167=AXIS2_PLACEMENT_3D('HOLE3',#168,#169,#170);
#168=CARTESIAN_POINT('HOLE3: LOCATION ',(0.000,17.33,0.000));
#169=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#170=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#171=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE3',#172);
#172=AXIS2_PLACEMENT_3D('HOLE3 DEPTH',#173,#174,#175);
#173=CARTESIAN_POINT('HOLE3: DEPTH ',(0.000,0.000,-13.750));
#174=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#175=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#176=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#177=CONICAL_HOLE_BOTTOM(49,$);

#178=AXIS2_PLACEMENT_3D('HOLE4',#179,#180,#181);
#179=CARTESIAN_POINT('HOLE4: LOCATION ',(0,-21.65,-32.5));
#180=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#181=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#182=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE5',#183);
#183=AXIS2_PLACEMENT_3D('HOLE4 DEPTH',#184,#185,#186);
#184=CARTESIAN_POINT('HOLE4: DEPTH ',(0.000,0.000,-3.000));
#185=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#186=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#187=TOLERANCED_LENGTH_MEASURE(6.000,#125);
#188=FLAT_HOLE_BOTTOM();

#189=AXIS2_PLACEMENT_3D('HOLE5',#190,#191,#192);
#190=CARTESIAN_POINT('HOLE5: LOCATION ',(18.75,-10.82,-32.5));
#191=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#192=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#193=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE5',#194);
#194=AXIS2_PLACEMENT_3D('HOLE5 DEPTH',#195,#196,#197);
#195=CARTESIAN_POINT('HOLE6: DEPTH ',(0.000,0.000,-3.000));
#196=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#197=DIRECTION(' AXIS ',(1.000,0.000,0.000));
```

```
#198=TOLERANCED_LENGTH_MEASURE(6.000,#125);
#199=FLAT_HOLE_BOTTOM();

#200=AXIS2_PLACEMENT_3D('PLACEMENT
CIRCULAR_CLOSED_PROFILE',#201,#202,#203);
#201=CARTESIAN_POINT('CIRCULAR_CLOSED_PROFILE: LOCATION
',(25.000,0.000,-28.00));
#202=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#203=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#250=ELEMENTARY_SURFACE('DEPTH SURFACE FOR
CIRCULAR_CLOSED_PROFILE',#251);
#251=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT
CIRCULAR_CLOSED_PROFILE',#252,#253,#254);
#252=CARTESIAN_POINT('CIRCULAR_CLOSED_PROFILE: DEPTH
',(0.000,0.000,-77.000));
#253=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#254=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#255=LINEAR_PATH('CIRCULAR_CLOSED_PROFILE',#256);
#256=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT
CIRCULAR_CLOSED_PROFILE',#257,#258,#259);
#257=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES SWEPT SHAPE:
DISTANCE',(0.000,0.000,-77.000));
#258=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#259=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));
#260=CIRCULAR_CLOSED_PROFILE($,#261);
#261=TOLERANCED_LENGTH_MEASURE(50.000,#125);

#400=AXIS2_PLACEMENT_3D('POCKET1',#401,#402,#403);
#401=CARTESIAN_POINT('POCKET1:LOCATION ',(-19.670,-11.350,-1.625));
#402=DIRECTION('AXIS ',(0.000,0.000,1.000));
#403=DIRECTION('REF_DIRECTION',(-1.000,0.000,0.000));
#404=ELEMENTARY_SURFACE('DEPTH SURFACE FOR POCKET1',#405);
#405=AXIS2_PLACEMENT_3D('POCKET1',#406,#407,#408);
#406=CARTESIAN_POINT('POCKET1:DEPTH ',(0.000,0.000,-3.000));
#407=DIRECTION('AXIS ',(0.000,0.000,1.000));
#408=DIRECTION('REF_DIRECTION',(1.000,0.000,0.000));
#409=PLANAR_POCKET_BOTTOM_CONDITION();
#410=TOLERANCED_LENGTH_MEASURE(2.50,#125);
#411=GENERAL_CLOSED_PROFILE($,#412);
#412=POLYLINE('CONTOUR OF POCKET1',(#413,#414,#415,#416,#413));
#413=CARTESIAN_POINT('P1',(0.000,0.000,0.000));
#414=CARTESIAN_POINT('P2',(0.000,10.500,0.000));
#415=CARTESIAN_POINT('P3',(-11.220,10.500,0.000));
#416=CARTESIAN_POINT('P4',(-11.220,0.000,0.000));

#417=AXIS2_PLACEMENT_3D('POCKET2',#418,#419,#420);
#418=CARTESIAN_POINT('POCKET2:LOCATION ',(19.670,11.350,-1.625));
#419=DIRECTION('AXIS ',(0.000,0.000,1.000));
```

```
#420=DIRECTION('REF_DIRECTION',(-1.000,0.000,0.000));
#421=ELEMENTARY_SURFACE('DEPTH SURFACE FOR POCKET2',#422);
#422=AXIS2_PLACEMENT_3D('POCKET2',#423,#424,#425);
#423=CARTESIAN_POINT('POCKET2:DEPTH ',(0.000,0.000,-3.000));
#424=DIRECTION('AXIS ',(0.000,0.000,1.000));
#425=DIRECTION('REF_DIRECTION',(1.000,0.000,0.000));
#426=PLANAR_POCKET_BOTTOM_CONDITION();
#427=TOLERANCED_LENGTH_MEASURE(2.50,#125);
#428=GENERAL_CLOSED_PROFILE($,#429);
#429=POLYLINE('CONTOUR OF POCKET2',(#430,#431,#432,#433,#430));
#430=CARTESIAN_POINT('P1',(0.000,0.000,0.000));
#431=CARTESIAN_POINT('P2',(0.000,10.500,0.000));
#432=CARTESIAN_POINT('P3',(-11.220,10.500,0.000));
#433=CARTESIAN_POINT('P4',(-11.220,0.000,0.000));

#300=MILLING_CUTTING_TOOL('FACE MILL 60MM',#306,(),$,$,$);
#301=MILLING_TECHNOLOGY(0.0166,.TCP.,$,50.0,$,$,$,$,$);
#302=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#303=PLUNGE_RAMP($,$);
#304=PLUNGE_RAMP($,$);
#305=UNIDIRECTIONAL($,$,$,$);
#306=FACEMILL(#307,$,$,$,$);
#307=MILLING_TOOL_DIMENSION(60.0,$,$,$,0.0,$,$);
#310=MILLING_CUTTING_TOOL('SLOT DRILL 12MM',#316,(),$,$,$);
#311=MILLING_TECHNOLOGY(0.0133,.TCP.,$,66.666,$,$,$,$,$);
#312=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#313=PLUNGE_RAMP($,$);
#314=PLUNGE_RAMP($,$);
#315=CONTOUR_PARALLEL($,$,$,$);
#316=ENDMILL(#317,$,$,$,$);
#317=MILLING_TOOL_DIMENSION(12.0,$,$,$,0.0,$,$);
#320=MILLING_CUTTING_TOOL('SLOT DRILL 6MM',#326,(),$,$,$);
#321=MILLING_TECHNOLOGY(0.0133,.TCP.,$,66.666,$,$,$,$,$);
#322=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#323=PLUNGE_RAMP($,$);
#324=PLUNGE_RAMP($,$);
#325=CONTOUR_PARALLEL($,$,$,$);
#326=ENDMILL(#327,$,$,$,$);
#327=MILLING_TOOL_DIMENSION(6.0,$,$,$,0.0,$,$);
#330=MILLING_CUTTING_TOOL('SPIRAL_DRILL_10MM',#333,(),$,$,$);
#331=MILLING_TECHNOLOGY(0.0133,.TCP.,$,66.666,$,$,$,$,$);
#332=MILLING_MACHINE_FUNCTIONS(.F.,$,$,$,$,(),$,$,$,());
#333=TWIST_DRILL(#334,2,.RIGHT.,.F.,0.840);
#234=MILLING_TOOL_DIMENSION(10.000,49.000,$,$,$,$,$);

ENDSEC;

END-ISO-10303-21;
```

12.8 STEP-NC code for Turn-Mill centre part 3

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('EXAMPLE OF NC PROGRAMME FOR TURN-MILLING: EXAMPLE
3.'),'1');
FILE_NAME('EXAMPLE3.STP',$,('ISO14649'),(''),'MEHRDAD
SAFAIEH','UNIVERSITY OF BATH', 'BATH');
FILE_SCHEMA(('MACHINING_SCHEMA','TURNING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;
DATA;

// WORKPIECE DEFINITION //

#1=WORKPIECE('SIMPLE WORKPIECE',$,$,$,$,#2,());
#2=RIGHT_CIRCULAR_CYLINDER('WORKPIECE PIECE', #3,65.0, 29.0);
#3=AXIS1_PLACEMENT('WORKPIECE PIECE PLACEMENT',#4,#5,#6);
#4=CARTESIAN_POINT('WORKPIECE PIECE: LOCATION
',(0.000,0.000,0.000));
#5=DIRECTION('AXIS ',(0.000,0.000,1.000));
#6=DIRECTION('REF_DIRECTION',(1.000,0.000,0.000));

// MANUFACTURING FEATURES //

#10=REVOLVED_FLAT('REVOLVED FLAT',#1,(#20,#21),#100,$,0.0,#104);
#11=GENERAL_OUTSIDE_PROFILE('NGON PROFILE WITH 6
SIDES',#1,(#26,#27),#109,#113,#118,#123);
#12=GENERAL_REVOLUTION('GENERAL_REVOLUTION',#1,(#22,#23),#126,$,26.2
2,#130);
#13=GROOVE('GROOVE 1',#1,(#30,#31),#200,#204,21.25,#205);
#14=ROUND_HOLE('HOLE1 CONICAL
BOTTOM',#1,(#25),#145,#149,#154,$,#155);
#15=ROUND_HOLE('HOLE2 CONICAL
BOTTOM',#1,(#25),#156,#160,#165,$,#166);
#16=ROUND_HOLE('HOLE3 CONICAL
BOTTOM',#1,(#25),#167,#171,#176,$,#177);
#17=ROUND_HOLE('HOLE4 FLAT BOTTOM',#1,(#24),#178,#182,#187,$,#188);
#18=ROUND_HOLE('HOLE5 FLAT BOTTOM',#1,(#24),#189,#193,#198,$,#199);
#19=CLOSED_POCKET('POCKET1',#1,(#28,#29),#215,#219,(),$,#224,$,#225,
#226);
#9=CLOSED_POCKET('POCKET2',#1,(#28,#29),#240,#244,(),$,#249,$,#250,#
251);


// OPERATIONS //

#20=FACING_ROUGH($,$,'ROUGH FACE',$,$,#280,#61,#60,$,$,$,0.500);
```

```
#21=FACING_FINISH($,$,'FINISH FACE',$,$,#280,#63,#60,$,$,$,0.0);
#22=CONTOURING_ROUGH($,$,'ROUGH GENERAL
REVOLUTION',$,$,#280,#61,#60,$,$,$,0.5);
#23=CONTOURING_FINISH($,$,'FINISH GENERAL
REVOLUTION',$,$,#280,#63,#60,$,$,$,0.0);
#24=DRILLING($,$,'DRILL HOLE FLAT
END',$,$,#293,#66,#65,$,$,$,$,$,$);
#25=DRILLING($,$,'DRILL HOLE CONICAL
END',$,$,#289,#66,#65,$,$,$,$,$,$);
#26=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGH
NGON6',$,$,#293,#66,#65,$,$,$,$,$,$,1.000,0.500);
#27=BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISH
NGON6',$,$,#293,#66,#65,$,$,$,$,$,$,$,$);
#28=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'ROUGH
POCKET',$,$,#320,#66,#65,$,$,$,$,$,$,1.000,0.500);
#29=BOTTOM_AND_SIDE_FINISH_MILLING($,$,'FINISH
POCKET',$,$,#320,#66,#65,$,$,$,$,$,$,$,$);
#30=GROOVING_ROUGH($,$,'ROUGH
GROOVE',$,$,#297,#61,#60,$,$,$,$,0.500);
#31=GROOVING_FINISH($,$,'FINISH
GROOVE',$,$,#297,#63,#60,$,$,$,$,0.0);

// PROJECT  //

#34=PROJECT('TURN-MILL EXAMPLE 3',#35,(#1),$,$,$);
#35=WORKPLAN('MAIN
WORKPLAN',(#38,#39,#40,#41,#42,#43,#44,#45,#46,#47,#49,#50,#51,#52,#
53,#54),$,#70,$);
#38=MACHINING_WORKINGSTEP('ROUGH FACE',#72,#10,#20,$);
#39=MACHINING_WORKINGSTEP('FINISH FACE',#72,#10,#21,$);
#40=MACHINING_WORKINGSTEP('ROUGH CONTOURING GENERAL
REVOLUTION',#72,#12,#22,$);
#41=MACHINING_WORKINGSTEP('FINISH CONTOURING GENERAL
REVOLUTION',#72,#12,#23,$);
#42=MACHINING_WORKINGSTEP('ROUGH BOTTOM AND SIDE MILLING NGON 6
SIDE',#72,#11,#26,$);
#43=MACHINING_WORKINGSTEP('FINISH BOTTOM AND SIDE MILLING NGON 6
SIDE',#72,#11,#27,$);
#44=MACHINING_WORKINGSTEP('DRILLING HOLE 1’,#72,#14,#25,$);
#45=MACHINING_WORKINGSTEP('DRILLING HOLE 2’,#72,#15,#25,$);
#46=MACHINING_WORKINGSTEP('DRILLING HOLE 3’,#72,#16,#25,$);
#47=MACHINING_WORKINGSTEP('DRILLING HOLE 4’,#72,#17,#24,$);
#48=MACHINING_WORKINGSTEP('DRILLING HOLE 5’,#72,#18,#24,$);
#49=MACHINING_WORKINGSTEP('ROUGH GROOVING',#72,#13,#30,$);
#50=MACHINING_WORKINGSTEP('FINISH GROOVING',#72,#13,#31,$);
#51=MACHINING_WORKINGSTEP('ROUGH POCKET 1',#72,#19,#28,$);
#52=MACHINING_WORKINGSTEP('FINISH POCKET 1',#72,#19,#28,$);
#53=MACHINING_WORKINGSTEP('ROUGH POCKET 2',#72,#9,#29,$);
```

```
#54=MACHINING_WORKINGSTEP('FINISH POCKET 2',#72,#9,#29,$);

#70=SETUP('SETUP 1',$,#71,(#72));
#71=PLANE('SECURITY PLANE',#73);
#72=WORKPIECE_SETUP(#1,#75,$,$,$);
#73=AXIS2_PLACEMENT_3D('SECURITY PLANE',#74,$,$);
#74=CARTESIAN_POINT('SECPLANE: LOCATION',(120.000,0.000,200.000));
#75=AXIS2_PLACEMENT_3D('WORKPIECE',#76,#77,#78);
#76=CARTESIAN_POINT('WORKPIECE: LOCATION',(0.000,0.000,0.000));
#77=DIRECTION('WORKPIECE: AXIS',(0.000,0.000,1.000));
#78=DIRECTION('WORKPIECE: REF_DIRECTION',(1.000,0.000,0.000));

// FUNCTIONS AND TECHNOLOGY //

#60=TURNING_MACHINE_FUNCTIONS(.T.,$,$,(),.F.,$,$,(),$,$,$);
#61=TURNING_TECHNOLOGY($,.TCP.,#62,0.300,.F.,.F.,.F.,$);
#62=CONST_SPINDLE_SPEED(500);
#63=TURNING_TECHNOLOGY($,.TCP.,#64,0.300,.F.,.F.,.F.,$);
#64=CONST_SPINDLE_SPEED(1000);
#65= MILLING_MACHINE_FUNCTIONS(.T.,$,$,.F.,$,(),.T.,$,$,());
#66= MILLING_TECHNOLOGY(0.030,.TCP.,#67,16.000,$,.F.,.F.,.F.,$);
#67=CONST_SPINDLE_SPEED(200);

// PLACEMENTS AND LENGTHS //

#100=AXIS2_PLACEMENT_3D('PLACEMENT REVOLVED FLAT',#101,#102,#103);
#101=CARTESIAN_POINT('REVOLVED FLAT: LOCATION
',(0.000,0.000,0.000));
#102=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#103=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#104=LINEAR_PROFILE('REVOLVED_FLAT_RADIUS',#105,29.000);
#105=AXIS2_PLACEMENT_3D('PLACEMENT',#106,#107,#108);
#106=CARTESIAN_POINT('LOCATION ',(0.000,0.000,0.000));
#107=DIRECTION(' AXIS ',(0.000,0.000,1.000));
#108=DIRECTION(' REF_DIRECTION',(1.000,0.000,0.000));

#109=AXIS2_PLACEMENT_3D('PLACEMENT NGON PROFILE WITH 6
SIDES',#110,#111,#112);
#110=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES: LOCATION
',(26.22,0.000,0.000));
#111=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#112=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#113=ELEMENTARY_SURFACE('DEPTH SURFACE FOR NGON PROFILE WITH 6
SIDES',#114);
#114=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 6
SIDES',#115,#116,#117);
#115=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES: DEPTH
',(0.000,0.000,-15.500));
```

```
#116=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#117=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#118=LINEAR_PATH('NGON PROFILE WITH 6 SIDES',#119);
#119=AXIS2_PLACEMENT_3D('DEPTH PLACEMENT NGON PROFILE WITH 6
SIDES',#120,#121,#122);
#120=CARTESIAN_POINT('NGON PROFILE WITH 6 SIDES SWEPT SHAPE:
DISTANCE',(26.220));
#121=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#122=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#123=NGON_PROFILE($,#124,6,.F.);
#124=TOLERANCED_LENGTH_MEASURE(26.22,#125);
#125=PLUS_MINUS_VALUE(0.100,0.100,3);

#126=AXIS2_PLACEMENT_3D('PLACEMENT
GENERAL_REVOLUTION',#127,#128,#129);
#127=CARTESIAN_POINT('GENERAL_REVOLUTION: LOCATION ',(26.22,0.000,-
15.5));
#128=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#129=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#130=GENERAL_PROFILE($,#131);
#131=POLYLINE('',(#132,#133));
#132=CARTESIAN_POINT('',(25.000,0.000, -15.5));
#133=CARTESIAN_POINT('',(25.000,0.000, -28.000));

#145=AXIS2_PLACEMENT_3D('HOLE1',#146,#147,#148);
#146=CARTESIAN_POINT('HOLE1: LOCATION ',(0.000,0.000,0.000));
#147=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#148=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#149=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE1',#150);
#150=AXIS2_PLACEMENT_3D('HOLE1 DEPTH',#151,#152,#153);
#151=CARTESIAN_POINT('HOLE1: DEPTH ',(0.000,0.000,-13.750));
#152=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#153=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#154=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#155=CONICAL_HOLE_BOTTOM(49,$);

#156=AXIS2_PLACEMENT_3D('HOLE2',#157,#158,#159);
#157=CARTESIAN_POINT('HOLE2: LOCATION ',(0.000,-17.33,0.000));
#158=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#159=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#160=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE2',#161);
#161=AXIS2_PLACEMENT_3D('HOLE2 DEPTH',#162,#163,#164);
#162=CARTESIAN_POINT('HOLE2: DEPTH ',(0.000,0.000,-13.750));
#163=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#164=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#165=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#166=CONICAL_HOLE_BOTTOM(49,$);
```

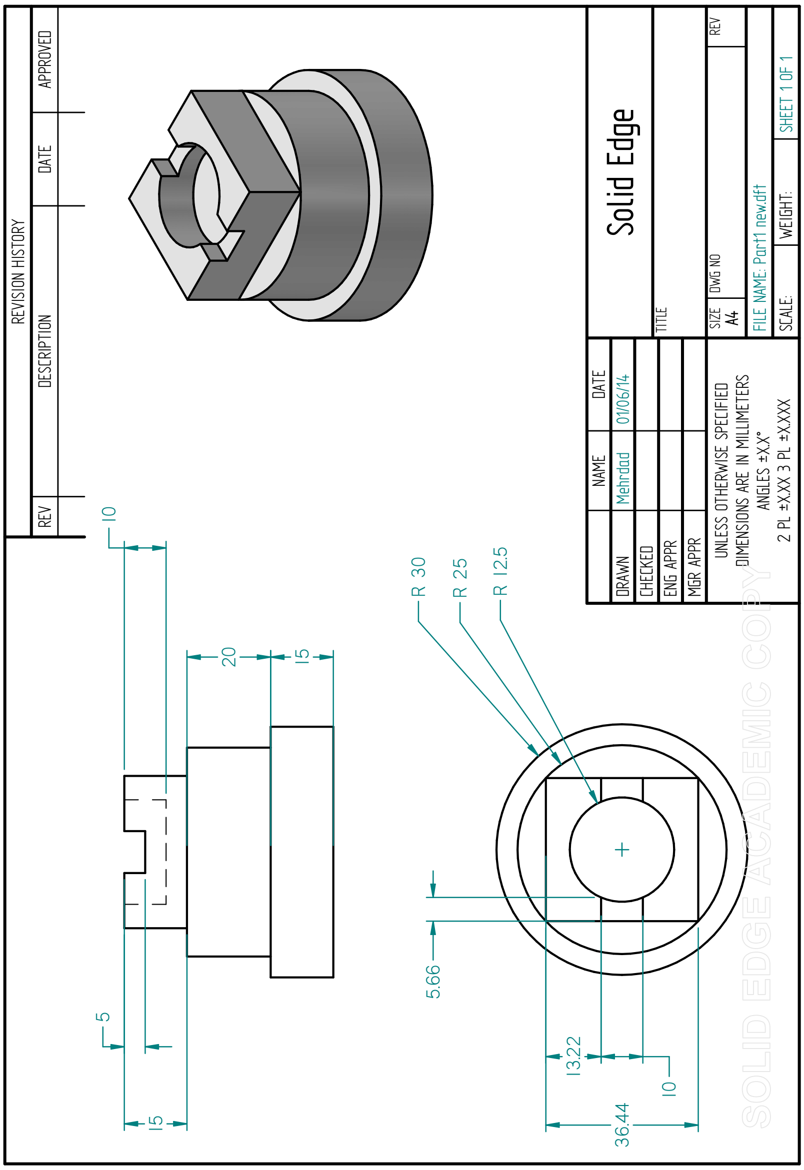
```
#167=AXIS2_PLACEMENT_3D('HOLE3',#168,#169,#170);
#168=CARTESIAN_POINT('HOLE3: LOCATION ',(0.000,17.33,0.000));
#169=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#170=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#171=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE3',#172);
#172=AXIS2_PLACEMENT_3D('HOLE3 DEPTH',#173,#174,#175);
#173=CARTESIAN_POINT('HOLE3: DEPTH ',(0.000,0.000,-13.750));
#174=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#175=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#176=TOLERANCED_LENGTH_MEASURE(10.000,#125);
#177=CONICAL_HOLE_BOTTOM(49,$);

#178=AXIS2_PLACEMENT_3D('HOLE4',#179,#180,#181);
#179=CARTESIAN_POINT('HOLE4: LOCATION ',(0,-21.65,-32.5));
#180=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#181=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#182=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE5',#183);
#183=AXIS2_PLACEMENT_3D('HOLE4 DEPTH',#184,#185,#186);
#184=CARTESIAN_POINT('HOLE4: DEPTH ',(0.000,0.000,-3.000));
#185=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#186=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#187=TOLERANCED_LENGTH_MEASURE(6.000,#125);
#188=FLAT_HOLE_BOTTOM();

#189=AXIS2_PLACEMENT_3D('HOLE5',#190,#191,#192);
#190=CARTESIAN_POINT('HOLE5: LOCATION ',(18.75,-10.82,-32.5));
#191=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#192=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#193=ELEMENTARY_SURFACE('DEPTH SURFACE FOR ROUND HOLE5',#194);
#194=AXIS2_PLACEMENT_3D('HOLE5 DEPTH',#195,#196,#197);
#195=CARTESIAN_POINT('HOLE6: DEPTH ',(0.000,0.000,-3.000));
#196=DIRECTION(' REF_DIRECTION ',(0.000,0.000,1.000));
#197=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#198=TOLERANCED_LENGTH_MEASURE(6.000,#125);
#199=FLAT_HOLE_BOTTOM();

#200=AXIS2_PLACEMENT_3D('PLACEMENT GROOVE',#201,#202,#203);
#201=CARTESIAN_POINT(' GROOVE : LOCATION ',(0.000,0.000, -28.000));
#202=DIRECTION(' AXIS ',(1.000,0.000,0.000));
#203=DIRECTION(' REF_DIRECTION',(0.000,0.000,1.000));
#204=DIRECTION(' MATERIAL_SIDE',(-1.000,0.000,0.000));
#205=SQUARE_U_PROFILE(#206,#207,0,#208,0);
#206=TOLERANCED_LENGTH_MEASURE(77.000,#125);
#207=TOLERANCED_LENGTH_MEASURE(0.000,#125);
#208=TOLERANCED_LENGTH_MEASURE(0.000,#125);

#215=AXIS2_PLACEMENT_3D('POCKET1',#216,#217,#218);
#216=CARTESIAN_POINT('POCKET1:LOCATION ',(-19.670,-11.350,-1.625));
```

```
#217=DIRECTION('AXIS ',(0.000,0.000,1.000));
#218=DIRECTION('REF_DIRECTION',(-1.000,0.000,0.000));
#219=ELEMENTARY_SURFACE('DEPTH SURFACE FOR POCKET1',#220);
#220=AXIS2_PLACEMENT_3D('POCKET1',#221,#222,#223);
#221=CARTESIAN_POINT('POCKET1:DEPTH ',(0.000,0.000,-3.000));
#222=DIRECTION('AXIS ',(0.000,0.000,1.000));
#223=DIRECTION('REF_DIRECTION',(1.000,0.000,0.000));
#224=PLANAR_POCKET_BOTTOM_CONDITION();
#225=TOLERANCED_LENGTH_MEASURE(2.50,#125);
#226=GENERAL_CLOSED_PROFILE($,#227);
#227=POLYLINE('CONTOUR OF POCKET1',(#228,#229,#230,#231,#228));
#228=CARTESIAN_POINT('P1',(0.000,0.000,0.000));
#229=CARTESIAN_POINT('P2',(0.000,10.500,0.000));
#230=CARTESIAN_POINT('P3',(-11.220,10.500,0.000));
#231=CARTESIAN_POINT('P4',(-11.220,0.000,0.000));

#240=AXIS2_PLACEMENT_3D('POCKET2',#241,#242,#243);
#241=CARTESIAN_POINT('POCKET2:LOCATION ',(19.670,11.350,-1.625));
#242=DIRECTION('AXIS ',(0.000,0.000,1.000));
#243=DIRECTION('REF_DIRECTION',(-1.000,0.000,0.000));
#244=ELEMENTARY_SURFACE('DEPTH SURFACE FOR POCKET2',#245);
#245=AXIS2_PLACEMENT_3D('POCKET2',#246,#247,#248);
#246=CARTESIAN_POINT('POCKET2:DEPTH ',(0.000,0.000,-3.000));
#247=DIRECTION('AXIS ',(0.000,0.000,1.000));
#248=DIRECTION('REF_DIRECTION',(1.000,0.000,0.000));
#449=PLANAR_POCKET_BOTTOM_CONDITION();
#250=TOLERANCED_LENGTH_MEASURE(2.50,#125);
#251=GENERAL_CLOSED_PROFILE($,#252);
#252=POLYLINE('CONTOUR OF POCKET2',(#253,#254,#255,#256,#253));
#253=CARTESIAN_POINT('P1',(0.000,0.000,0.000));
#254=CARTESIAN_POINT('P2',(0.000,10.500,0.000));
#255=CARTESIAN_POINT('P3',(-11.220,10.500,0.000));
#256=CARTESIAN_POINT('P4',(-11.220,0.000,0.000));


// TOOLS //

#280=TURNING_MACHINE_TOOL('',#281,(#283),120,40,$);
#281=GENERAL_TURNING_TOOL(#282,.LEFT.,40,60,.CW.);
#282=TOOL_DIMENSION($,$,$,$,25,5,7,3,5,0.5,$);
#283=CUTTING_COMPONENT(0.000000,$,$,$,$);

#289=
MILLING_CUTTING_TOOL('SPIRAL_DRILL_10MM',#290,(#292),90.000,$,$);
#290= TWIST_DRILL(#291,2,.RIGHT.,.F.,0.840);
#291=
MILLING_TOOL_DIMENSION(10.000,31.000,0.100,45.000,2.000,5.000,8.000)
;
```

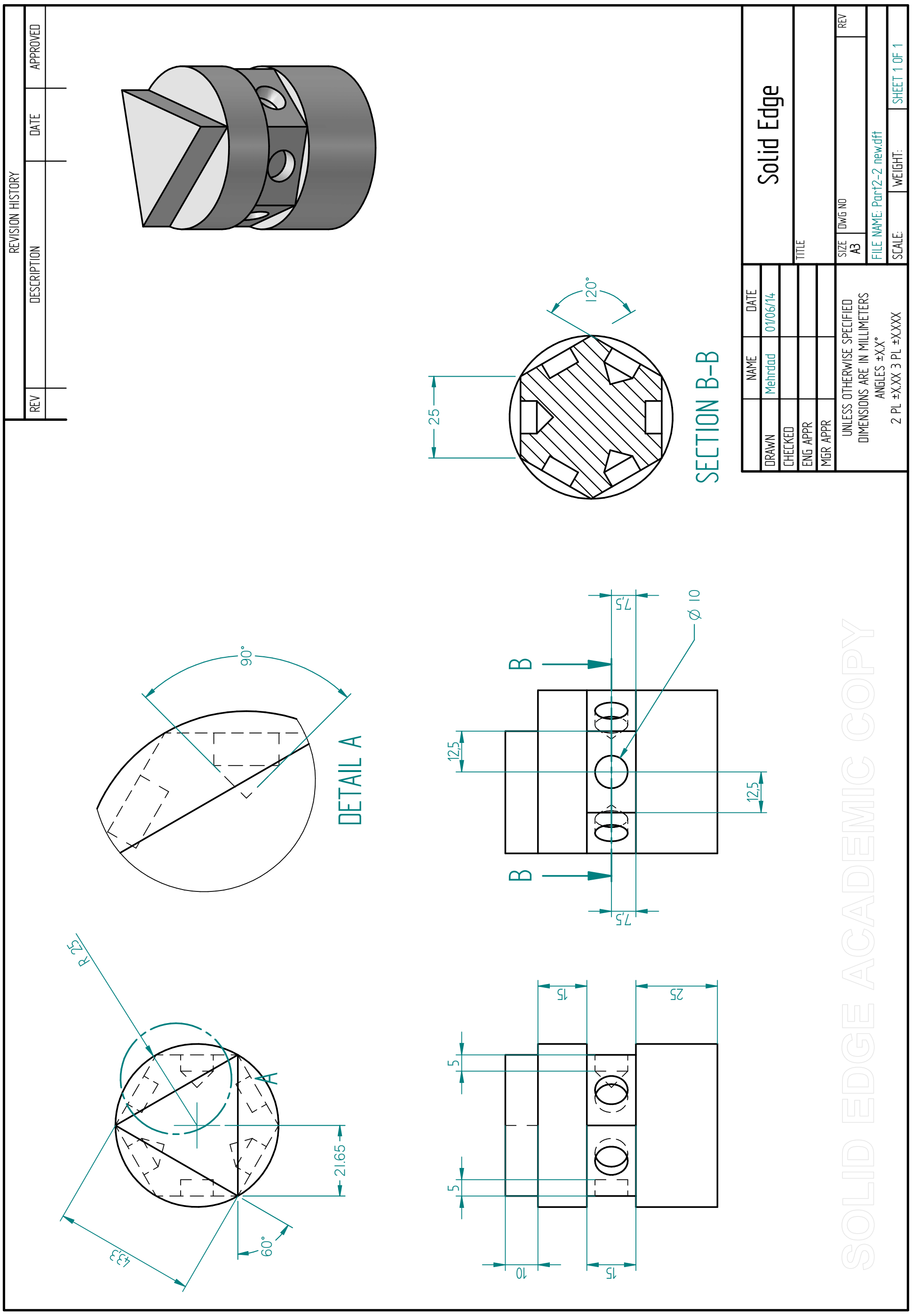
```
#292= CUTTING_COMPONENT(90.000,$,$,$,$);

#293= MILLING_CUTTING_TOOL('ENDMILL 10MM',#294,(#296),100.000,$,$);
#294= ENDMILL(#295,6,.RIGHT.,.F.,$);
#295= MILLING_TOOL_DIMENSION(10.000,$,$,$,$,$,$);
#296= CUTTING_COMPONENT(100.000,$,$,$,$);

#297=TURNING_MACHINE_TOOL('',#298,(#300),120,40,$);
#298=GROOVING_TURNING_TOOL(#299,.LEFT.,40,60,.CW.,10.0, $ );
#299=TOOL_DIMENSION($,$,$,$,$,$,$,$,$,0.5,$);
#300=CUTTING_COMPONENT(40.000,$,$,$,$);

#320=MILLING_CUTTING_TOOL('SLOT DRILL 6MM',#326,(),$,$,$);
#326=ENDMILL(#327,$,$,$,$);
#327=MILLING_TOOL_DIMENSION(6.0,$,$,$,0.0,$,$);

ENDSEC;
END-ISO-10303-21;
```

13 Appendix C. Test parts geometry

13.1 Part 1

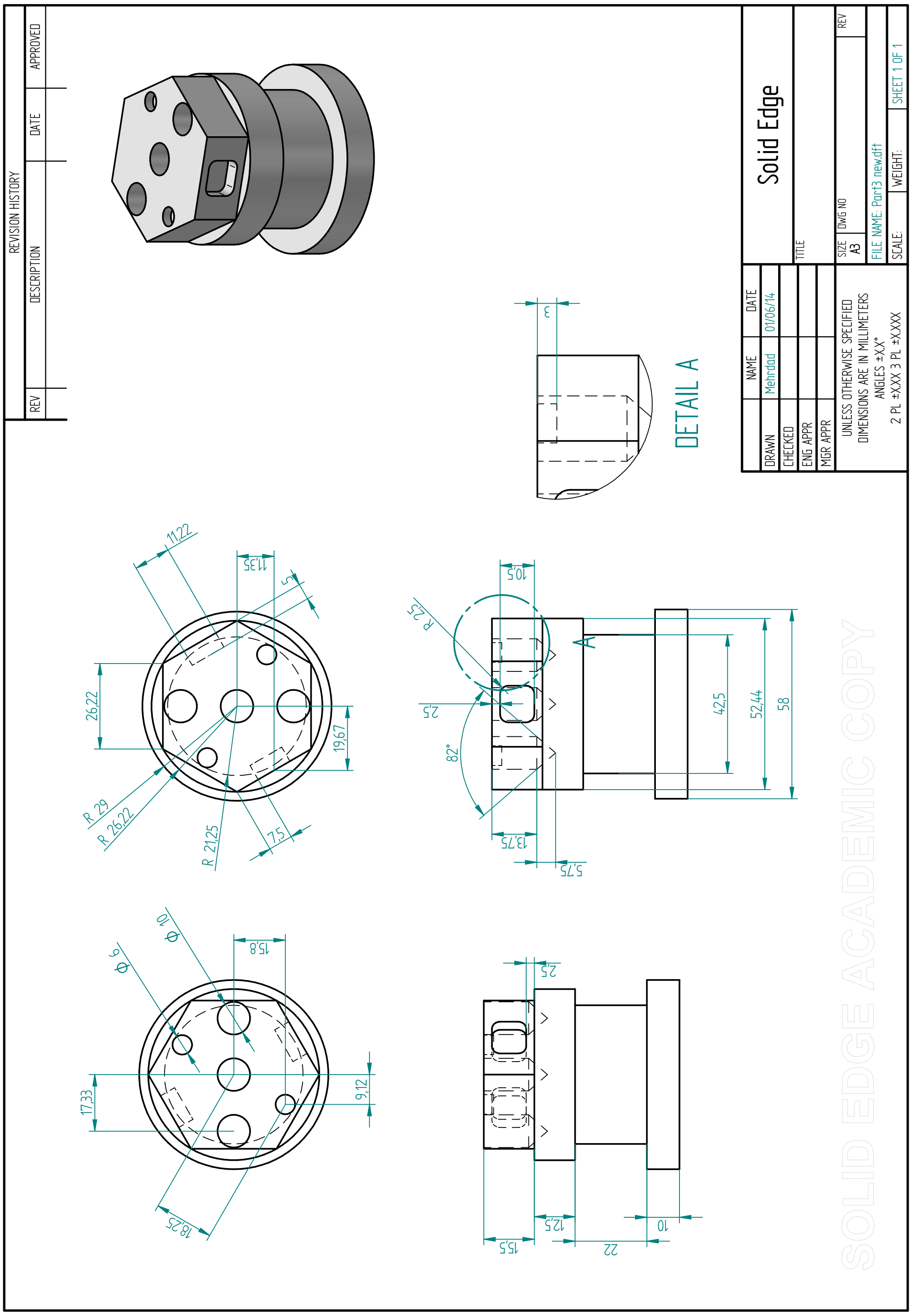


13.2 Part 2



REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
SECTION B-B
DETAIL A
120°
90°
60°
25
12,5
7,5
Ø 10
R 25
43,3
21,65
15
10
5
B
A
Solid Edge
TITLE
DRAWN
CHECKED
ENG APPR
MGR APPR
NAME
DATE
Mehrdad
01/06/14
UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX
SIZE
A3
DWG NO
REV
FILE NAME: Part2-2 new.dft
SCALE:
WEIGHT:
SHEET 1 OF 1
SOLID EDGE ACADEMIC COPY

13.3 Part 3


REVISION HISTORY
REV
DESCRIPTION
DATE
APPROVED
Solid Edge
DETAIL A
DRAWN
CHECKED
ENG APPR
MGR APPR
NAME
DATE
Mehrdad
01/06/14
UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN MILLIMETERS
ANGLES ±X.X°
2 PL ±X.XX 3 PL ±X.XXX
TITLE
SIZE
A3
DWG NO
REV
FILE NAME: Part3 new.dft
SCALE:
WEIGHT:
SHEET 1 OF 1
SOLID EDGE ACADEMIC COPY